APPENDIX A

ACRONYMS AND ABBREVIATIONS

APPENDIX A – ACRONYMS AND ABBREVIATIONS

AMC	Air Mobility Command
ASCII	American Standard Code for Information Interchange
BCD	Binary-Coded Decimal
BPI	Bits Per Inch
BROUTER	Bridging Router
CDAY	Commencement Day
CDRL	Contract Data Requirements List
CIN	Cargo Increment Number
COA	Course of Action
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
DART	Dynamic Analysis and Replanning Tool
DBA	Database Administrator
DBSE	Differential SCSI-2 Buffered Ethernet
DISA	Defense Information Systems Agency
DISN	Defense Information Systems Network
DOD	Department of Defense
DPS	Data Processing System
DRAM	Dynamic Random Access Memory
DSN	Defense Switched Network
EAD	Earliest Arrival Date
FDBM	Functional Database Manager
FM	Functional Manager
GBYTE	Giga-Byte
GCCS	Global Command and Control System
GEO	Geographic Locations File
GEOFILE	Geographic Locations File
GEOLOC	Geographic Locations
GID	Group Identifier
GUI	Graphical User Interface
ICG	ORACLE7 for Sun SPARC Solaris 2.3 Installation and Configuration Guide
ID	Identifier
IDS	Integrated Data Store
IMS	Information Management Subsystem
I/O	Input/Output
IP	Internet Protocol

JCL	Job Control Language
JES	JOPES Executive Subsystem
JFAST	Joint Flow and Analysis System for Transportation
JOPES	Joint Operation Planning and Execution System
JPEC	Joint Planning and Execution Community
Kbps	Kilo-Bits per Second
KB	Kilobyte
LAD	Latest Arrival Date
LAN	Local Area Network
LFF	Logistics Factors File
LHOUR	Launch Hour
LOGSAFE	Logistics Sustainment Analysis and Feasibility Estimator
MB	Megabyte
MAJCOM	Major Command
MBIT	Megabit
Mbps	Megabits Per Second
MHz	Megahertz
MQH	Memory Queue Handler
MSC	Military Sealift Command
MTMC	Military Traffic Management Command
OPLAN	Operation Plan
PC	Personal Computer
PID	Process Identifier
PIF	Problem Indicator Flags
PIN	Personnel Increment Number
POD	Port of Debarkation
POE	Port of Embarkation
QIC	Quarter Inch Cartridge
RAM	Random Access Memory
RDBMS	Relational Database Management System
RFM	Reference File Manager
SARG	ORACLE7 Server for Unix Administrator’s Reference Guide
SBSE	SCSI Buffered Ethernet
SCSI	Small Computer System Interface
SID	ORACLE7 System Identification
S&M	Scheduling and Movement
SNUMB	Sequence Number
SOP	Standard Operating Procedure

SRA	Systems Research and Applications Corporation
SRF	Summary Reference File
SQL	Sequential Query Language
SSF	Schedule Status Flags
SunOS	Sun (Corporation) Operating System
TA	Technical Administrator
TARG	ORACLE7 Tools for Unix Administrator's Reference Guide
TCC	Transportation Component Command
TCP	Transmission Control Protocol
TDBM	Technical Database Manager
TDS	Transaction Distribution Services
TIP	Technology Insertion Project
TP	Transaction Processor
TPFDD	Time-Phased Force and Deployment Data
TUCHA	Type Unit Characteristics Data
UID	User Identifier
ULN	Unit Line Number
USTRANSCOM	United States Transportation Command
UMC	User Master Catalog
UTC	Unit Type Code
WAN	Wide Area Network
WIS	WWMCCS Information System
WWMCCS	Worldwide Military Command and Control System
WWS	WIS Workstations
XTP	External Transaction Processor

APPENDIX B

ADMINISTRATION FOR IMS/RFM

APPENDIX B — ADMINISTRATION FOR IMS/RFM

Information Management System (IMS) and Reference File Manager (RFM) must be configured for the specific set of applications which they support. This section identifies the tools and files used to administer IMS and RFM.

B.1 IMS/RFM APPLICATION ADMINISTRATION

IMS/RFM administration primarily involves using the IMS Administration Tool and the RFM Administration Tool to configure IMS and RFM, respectively, with the applications and files that they must support. IMS must be configured with the set of applications which are TPFDD sources and destinations. RFM must be configured with the set of applications which are destinations for the defined standard reference files.

An environment configuration file is also necessary for IMS and RFM. This file is called IMS_RFM.env and is used to specify information about the environment in which IMS/RFM operates.

The IMS_RFM.env file, the IMS Administration Tool, and the RFM Administration Tool are described in the following sections.

B.1.1 The IMS/RFM Segment

The IMS/RFM software resides within the IMS/RFM segment. In the GCCS environment, the segment is typically located in the "/h/IMS_RFM" directory. These directories are created within the segment for the execution of IMS/RFM:

- **app-defaults**

  This directory contains the Screen Machine file used to define the resources for the IMS/RFM screens. The file is called IMSRFM.

- **bin**

  This directory contains the executables for IMS and RFM. Both Ada executables and Unix shell scripts are stored here. This directory also holds the ims_apps file (described later).

- **bitmaps**

  This directory holds the bitmaps displayed during IMS/RFM execution.

- **files**

  This directory holds the IMS/RFM help files, as well as the refapp_info file (described later).

- **imsdata**

  This directory is the holding area for TPFDDs stored within IMS and for reference files stored within RFM. The reference files are stored in the "refs" directory underneath of imsdata. The imsdata directory also contains a file which holds information about the TPFDDs stored in IMS. The default name of this file is "tpfdd_info1", however the name may be tailored in the IMS_RFM.env file (described later).

- **refapp_info**

  This directory holds the Unix shell scripts used to create reference files and import them into the various applications.

- **Scripts**

  This directory holds the startup scripts used to launch IMS, RFM, the IMS Admin Tool and the RFM Admin Tool.

B.1.2 Launching IMS, RFM and the Admin Tools

IMS, RFM, the IMS Admin Tool and the RFM Admin Tool are invoked via Unix C-shell scripts. The scripts reside in the "Scripts" directory and are named as follows:

- start_ims
- start_ims_admin
- start_rfm
- start_rfm_admin

By default, each of the scripts, and therefore each of the tools, may be invoked by the installer and/or anyone in the "gccs" Unix group. To limit access to the tools (for example, to the 'JADMIN' group), change the group ID of the file owner.

B.1.3 IMS_RFM.env

IMS and RFM have been designed to operate with minimal knowledge of the execution environment. Specifically, IMS/RFM do not have hard-coded references to directories. In order to achieve this flexibility, the Ada software and Unix shell scripts which comprise IMS/RFM make reference to Unix environment variables. All such variables are collected in the IMS_RFM.env file. The environment variables are explicitly set in this file, which is a Unix C-shell script. Upon execution of IMS, RFM, the IMS Admin Tool or the RFM Admin Tool this file is sourced in order to establish the values of the environment variables.

The IMS_RFM.env file establishes the following configuration information:

- Directory name for IMS/RFM Segment (typically /h/IMS_RFM)
- IMS/RFM directory variables (for IMS/RFM work space)
- S&M directory variables
- Dynamic Analysis and Replanning Tool (DART) directory variables
- Joint Flow and Analysis System for Transportation (JFAST) directory variables.

In addition to setting environment variables for the use of IMS/RFM, sourcing this file also performs the following activities:

- Sources DART's environment variable file (dart.env)
- Sources the Oracle environment variable file (coraenv)
- Appends paths for Oracle, IMS/RFM and "/usr/ccs" to the user's path.

The IMS_RFM.env file is stored in the "bin" directory. It is delivered with default data that is not expected to change. However, if the environment changes, this file may be tailored to provide the new environment information.

B.1.4 IMS Administration Tool

The IMS Admin Tool is used to configure TPFDD sources and destinations for IMS. Using this tool, each application is specified as a TPFDD source, a TPFDD destination or both. For each application, up to three software programs are identified to perform the following activities:

- Get a list of TPFDDs from the application (if the application is a TPFDD source). This program must return a list of OPLANs for which a TPFDD file can be retrieved. The list of OPLANs will be displayed to the user in a list box.

- Extract a TPFDD (export) from the application (if the application is a TPFDD source). This program implements the moving of a TPFDD file from an application into IMS. This program has application-specific knowledge of where the TPFDD resides within the application. By convention, this program is passed two parameters: The OPLAN number and the name of a temporary file to be used during the export.

- Import a TPFDD into the application (if the application is a TPFDD destination). This program implements the moving of a TPFDD file from IMS to an application. It has application-specific knowledge of how to import a TPFDD into the application. By convention, this program is passed two parameters: The filename of the TPFDD file and the OPLAN number.

The information generated by running the IMS Admin Tool is stored in a binary file called "ims_apps". ims_apps is stored in the "bin" directory and is read by IMS when it is invoked. For each application, this file stores the following information:

- **Application Name** - The name of the application importing or exporting TPFDDs.

- **Can Application Import TPFDDs** - Indicates whether a TPFDD can be imported into the application.

- **Reset IMS Upload Date** - Indicates whether the date the TPFDD was imported to an application should be modified.

- **Run Import In Background** - Indicates whether the import program should be run as a background process or in the foreground.

- **Run Import In Xterm** - Indicates whether the import program should be run in an xterm window or not.

- **TPFDD Import File** - The name of the program which imports a TPFDD into the application.

- **TPFDD Import Path** - The directory which holds the program which imports a TPFDD into the application.

- **Can Application Export TPFDDs** - Indicates whether a TPFDD can be exported from the application.

- **Reset IMS Download Date** - Indicates whether the date the TPFDD was downloaded should be modified.

- **Run Export In Xterm** - Indicates whether the export program should be run in an xterm window or not.

- **TPFDD Export File** - The name of the program which exports a TPFDD from the application.

- **TPFDD Export Path** - The directory which holds the program which exports a TPFDD from the application.

- **Any Interface Defined To Get List Of TPFDDs** - Indicates whether or not a program exists to provide a list of TPFDDs to the user. Yes means such a program exists. No means no program exists (the user will need to explicitly type the plan number of the desired TPFDD).

- **Interface File** - The name of the program which obtains a list of TPFDDs from the selected application.

- **Interface Path** - The directory which holds the program which obtains a list of TPFDDs from the selected application.

The following table shows the default information provided in the ims_apps file. In the event this information needs to be re-entered/updated, launch the IMS Admin Tool and click on "Add New Application". Enter the required data and then click on "save". **Note**: ***GCCS_HOME*** should be replaced by the directory location of the IMS/RFM Segment (typically "/h/IMS_RFM").

Table B-1: IMS Administration Tool Default Configuration.

IMS ADMINISTRATION TOOL DEFAULT CONFIGURATION	
INSTRUCTIONS	**ANSWERS**
IMS ADMIN TOOL FOR DART	
Application Name	DART
Can Application Import TPFDDs	YES
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	YES
TPFDD Import File	dart_load.csh
TPFDD Import Path	***GCCS_HOME***/bin
Can Application Export TPFDDs	YES
Reset IMS Download Date	NO
Run Export In Xterm	YES
TPFDD Export File	dump_tpfdd
TPFDD Export Path	/h/DART/data/dart/ Version_Current/scripts
Any Interface Defined To Get List Of TPFDDs	YES
Interface File	dart_files.csh
Interface Path	***GCCS_HOME***/bin
IMS ADMIN TOOL FOR EXTERNAL TPFDD FILE	
Application Name	External
Can Application Import TPFDDs	YES

IMS ADMINISTRATION TOOL DEFAULT CONFIGURATION	
INSTRUCTIONS	**ANSWERS**
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	YES
TPFDD Import File	b8_import.csh
TPFDD Import Path	***GCCS_HOME***/bin
Can Application Export TPFDDs	YES
Reset IMS Download Date	NO
Run Export In Xterm	YES
TPFDD Export File	b8_export.csh
TPFDD Export Path	***GCCS_HOME***/bin
Any Interface Defined To Get List Of TPFDDs	YES
Interface File	b8_files.csh
Interface Path	***GCCS_HOME***/bin
IMS ADMIN TOOL FOR GCCS JOPES DATABASE	
Application Name	GCCS JOPES DB
Can Application Import TPFDDs	NO
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	NO
TPFDD Import File	(blank)
TPFDD Import Path	(blank)
Can Application Export TPFDDs	YES
Reset IMS Download Date	NO
Run Export In Xterm	NO
TPFDD Export File	create_tpfdd.exe
TPFDD Export Path	***GCCS_HOME***/bin
Any Interface Defined to Get List Of TPFDDs	YES
Interface File	core_plan_list.csh
Interface Path	***GCCS_HOME***/bin

IMS ADMINISTRATION TOOL DEFAULT CONFIGURATION	
INSTRUCTIONS	**ANSWERS**
IMS ADMIN TOOL FOR JFAST	
Application Name	JFAST
Can Application Import TPFDDs	YES
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	YES
TPFDD Import File	jfast_load.csh
TPFDD Import Path	***GCCS_HOME***/bin
Can Application Export TPFDDs	NO
Reset IMS Download Date	NO
Run Export In Xterm	NO
TPFDD Export File	(blank)
TPFDD Export Path	(blank)
Any Interface Defined To Get List Of TPFDDs	NO
Interface File	(blank)
Interface Path	(blank)
IMS ADMIN TOOL FOR TRANSACTIONS	
Application Name	Transactions
Can Application Import TPFDDs	NO
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	NO
TPFDD Import File	(blank)
TPFDD Import Path	(blank)
Can Application Export TPFDDs	YES
Reset IMS Download Date	NO
Run Export in Xterm	YES
TPFDD Export File	dartx_export.csh
TPFDD Export Path	***GCCS_HOME***/bin

IMS ADMINISTRATION TOOL DEFAULT CONFIGURATION	
INSTRUCTIONS	**ANSWERS**
Any Interface Defined To Get List Of TPFDDs	YES
Interface File	dartx_files.csh
Interface Path	***GCCS_HOME***/bin
IMS ADMIN TOOL FOR XTP	
Application Name	XTP
Can Application Import TPFDDs	YES
Reset IMS Upload Date	NO
Run Import In Background	NO
Run Import In Xterm	YES
TPFDD Import File	xaction_into_xtp.csh
TPFDD Import Path	***GCCS_HOME***/bin
Can Application Export TPFDDs	NO
Reset IMS Download Date	NO
Run Export In Xterm	NO
TPFDD Export File	(blank)
TPFDD Export Path	(blank)
Any Interface Defined To Get List Of TPFDDs	NO
Interface File	(blank)
Interface Path	(blank)

B.1.5 RFM Admin Tool

The RFM Administration Tool is used to configure reference file destinations for RFM. The standard reference files are extracted from the JOPES Core Database. Using this tool, destination applications for each reference file are specified. For each reference file, one or more destination applications are specified and software programs are identified to load the reference file into the application.

The information generated by running this tool is stored in a binary file called “refapp_info”. refapp_info is stored in the “files” directory and is read by RFM when it is invoked. For each reference file, this file stores the following information:

- **Reference File** - The name of the reference file.
- **Reference File Name -** The Unix file name to be used for the extracted reference file.

- **Reference File Path** - The directory in which the extracted reference file will be stored within RFM.

- **Update Script** - The program used to extract the reference file from the JOPES Core Database and save in flat-file, ASCII format.

- **Date Offset** - The starting position of the date within the first record of the reference file. For some reference files, the date of the reference file is stored within the first, or header, record of the file. This is the reference file date and is the date displayed to the RFM user. If the date is not stored in the first record of the reference file, -1 should be entered here. In this case, the RFM user will see the date that the reference file was extracted.

- **App. Name** - The name of the application into which the reference file can be imported.

- **Machine Name** - Obsolete field. Leave blank.

- **Load File** - The program which loads the specified reference file into the specified application. The name of this program usually takes the form "<ref_file>_into_<application>.csh".

The following table shows the default information provided in the refapp_info file. In the event this information needs to be re-entered/updated, launch the RFM Administration Tool and follow these directions to enter all the default information:

For the first row for a given reference file:

- Enter the data shown in the first row listed for the reference file (e.g., the ASSETS row)
- Click on Save.

This adds the reference file and the application indicated in the first row for the reference file (e.g., DART).

Then, for each additional row for the same reference file:

- Click on Add New Application
- Enter the data listed in the last three columns of the row
- Click on Save.

This adds information for each application which can import the specified reference file.

Preform the first two steps once for each reference file. Perform the last three steps for each application that can import that reference file.

Note: ***GCCS_HOME*** should be replaced by the directory location of the IMS/RFM Segment (typically "/h/IMS_RFM").

Table B-2: Reference File Manager Administration Tool Default Configuration.

REFERENCE FILE MANAGER ADMINISTRATION TOOL DEFAULT CONFIGURATION							
Reference File	**Reference File Name**	**Reference File Path**	**Update Script**	**Date Offset**	**App. Name**	**Machine Name**	**Load File**
ASSETS	ASSETS.DAT	***GCCS_HOME***/ims data/refs	extract_assets.csh	-1	DART	(blank)	assets_into_dart.csh
ASSETS	ASSETS.DAT	***GCCS_HOME***/ims data/refs	extract_assets.csh	-1	JFAST	(blank)	assets_into_jfast.csh
CHSTR	CHSTR.DAT	***GCCS_HOME***/ims data/refs	extract_chstr.csh	-1	DART	(blank)	chstr_into_dart.csh
CHSTR	CHSTR.DAT	***GCCS_HOME***/ims data/refs	extract_chstr.csh	-1	JFAST	(blank)	chstr_into_jfast.csh
GEO	geofile.dat	***GCCS_HOME***/ims data/refs	extract_geo.csh	13	DART	(blank)	geo_into_dart.csh
GEO	geofile.dat	***GCCS_HOME***/ims data/refs	extract_geo.csh	13	JFAST	(blank)	geo_into_jfast.csh
GEO	geofile.dat	***GCCS_HOME***/ims data/refs	extract_geo.csh	13	MEPES	(blank)	geo_into_mepes.csh
TUCHA	TUCHA.DAT	***GCCS_HOME***/ims data/refs	extract_tucha.csh	19	DART	(blank)	tucha_into_dart.csh
TUCHA	TUCHA.DAT	***GCCS_HOME***/ims data/refs	extract_tucha.csh	19	JFAST	(blank)	tucha_into_jfast.csh
TUCHA	TUCHA.DAT	***GCCS_HOME***/ims data/refs	extract_tucha.csh	19	MEPES	(blank)	tucha_into_mepes.csh
UI	UIfile.load	***GCCS_HOME***/ims data/refs	extract_ui.csh	-1	DART	(blank)	ui_into_dart.csh

APPENDIX C

GCCS C/S DATABASE

BACKUP AND RECOVERY GUIDE (REVISION)

SECTION C.1 — BACKUP AND RECOVERY OVERVIEW

C.1.1 OVERVIEW

This guide provides reference information for the JOPES site administrators. It provides detailed procedures for use of backup and recovery scripts, and general guidance for the most common situations a site administrator may encounter. It is intended to complement Unix and ORACLE backup and recovery reference material. To get started quickly (recommended for experienced administrators only) refer to Paragraph C.2.1 Backup and Recovery Menu: Backup Options.

Users may reproduce this document as necessary.

C.1.2 REFERENCED DOCUMENTS

The following references apply to this document:

- *ORACLE for Sun SPARC Solaris 2.3 Installation and Configuration Guide (ICG)*, Release 7.0.13, Part Number A11854-1. Oracle Corporation, 12 Jul 93.

- *ORACLE Tools for Unix Administrator's Reference Guide (TARG) for 6.0 and 7.0*, Part Number A10323-1. Oracle Corporation, 7 May 93.

- *ORACLE7 Server Administrator's Guide*, Part Number 6694-70-1292. Oracle Corporation, Dec 92.

- *ORACLE7 Server Concepts Manual*, Part Number 6693-70-1292. Oracle Corporation, Dec 92.

- *ORACLE7 Server for Unix Administrator's Reference Guide (SARG)*, Release 7.0.13, Part Number A10324-1. Oracle Corporation, 15 Jun 93.

- *ORACLE7 Server Messages and Codes Manual*, Part Number 3605-70-1292. Oracle Corporation, Dec 92.

- *ORACLE7 Server Utilities User's Guide*, Part Number 3602-70-1292. Oracle Corporation, Dec 92.

- *SunOS 5.1 Reference Manual, User Commands (A-M)*, Revision A, Part Number 801-2840-10. Sun Microsystems, Inc., Mountain View, CA, Dec 92.

- *SunOS 5.1 Reference Manual, User Commands (A-M)*, Revision A, Part Number 801-2840-10. Sun Microsystems, Inc., Mountain View, CA, Dec 92.

- *JOPES Technical Database Manager's (TDBM) Handbook*, TD 18-64. Defense Systems Support Organization (DSSO), 16 Aug 93.

- *JOPES Functional Data Base Manager (FDBM) Users Manual - Volume 4*, TD 18-14-1. Defense Systems Support Organization, 17 Aug 93.

C.1.3 DOCUMENT NAMING CONVENTIONS

The following naming conventions apply throughout the document:

- UPPER CASE denotes menu screen titles and menu screen selections.

- **"bold within quotations"** represents messages that appear on the screen.

- **DOUBLE UNDERLINED BOLD** denotes tape label names.

- **UPPER CASE BOLD** represents ORACLE tablespaces, tables, and views. It is also used to represent mainframe files.

- **lower case bold** represents Unix files, modules, and all other programs. It is also used to identify important terminology throughout the document.

- *UPPER CASE ITALIC* represents ORACLE commands and all other commands that must be entered in upper case.

- *lower case italic* represents Unix commands and all other commands that must be entered in lower case text.

- <*bracketed italic*> represents commands entered by the user on the command-line.

- *UNDERLINED italic* represents variables and arguments. Upper and lower case represent the appearances in common use.

C.1.4 BACKUP AND RECOVERY CONCEPTS

The following paragraphs provide information that will help you understand some of the concepts upon which the GCCS backup and recovery software is based. The following paragraphs are not intended to replace the ORACLE7 Server Concepts Manual.

C.1.4.1 Databases, Tablespaces, and Data Files

A database is divided into logical storage units called **tablespaces**. A tablespace is used to group related logical structures together. Logical structures are database objects such as tables and views. **Data files** on disk are used to store all of the database data. A data file can be associated with only one database. A small example database is presented in Figure C-1 to illustrate the relationship among databases, tablespaces, and data files.

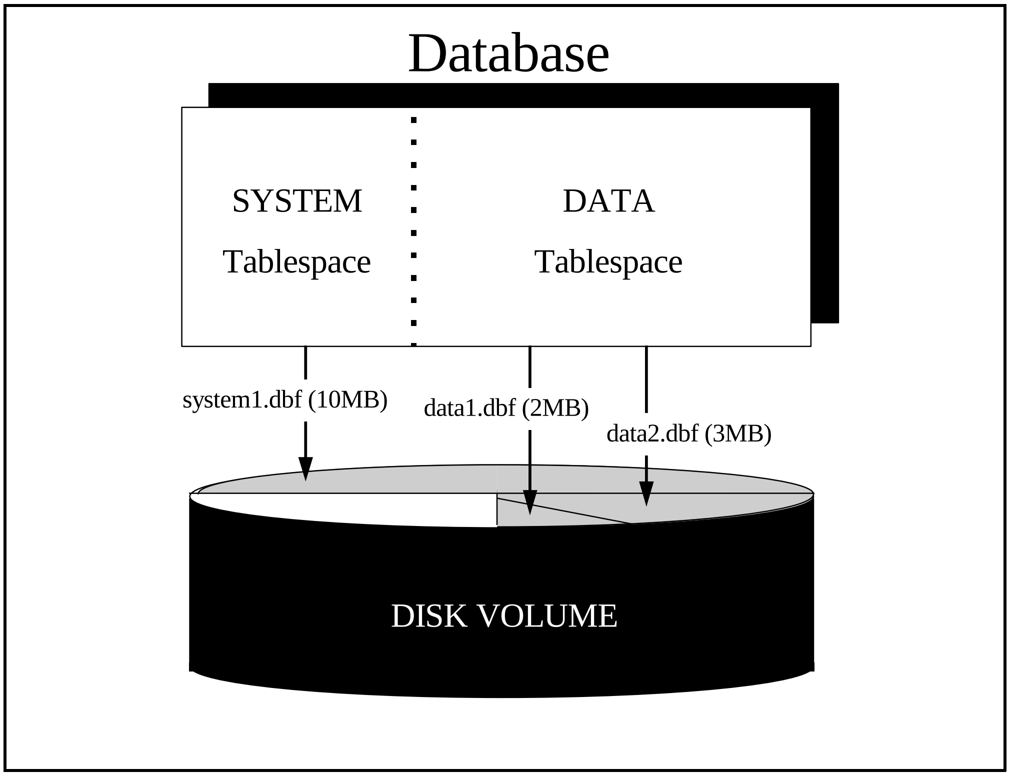


Figure C-1: Databases, Tablespaces, and Data Files

Figure C-1 illustrates:

- Each database is logically divided into one or more tablespaces (The example database is divided into two tablespaces; **SYSTEM** and **DATA**).

- One or more data files are explicitly created for each tablespace to physically store the data of all logical structures in a tablespace (The data for the example tablespace **SYSTEM** is physically stored in the data file **system1.dbf**. The data for the example tablespace **DATA** is physically stored in the data files **data1.dbf** and **data2.dbf**).

- The combined size of a tablespace's data files is the total storage capacity of the tablespace (The example tablespace **SYSTEM** has a 10 megabyte (MB) storage capacity and the example tablespace **DATA** has a 5 MB storage capacity).

- The combined storage capacity of a database's tablespaces is the total storage capacity of the database. (The example database has a 15 MB storage capacity).

For more detailed information on databases, tablespaces, and data files refer to the ORACLE7 Server Concepts Manual.

C.1.4.2 Archived Redo Log Files

The database **redo log files** contain structural and data value changes made to the database. **Online redo log files** contain recent changes. The GCCS database has 10 online redo log files. When an online redo log file becomes full, it must eventually be overwritten with new changes. However, before an online redo log file can be overwritten, its contents must be saved in an **archived redo log file**. Archived redo log files are stored in the archive directory **/oracle/smback/arch**. The term "redo log" throughout this document generally refers to the archived redo log file. If the archive directory becomes full, the database will cease operations until some of the archived redo log files are removed. The backup and recovery software allows you to move the archived redo log files to tape.

For more detailed information on the use of backup and recovery software to manage redo logs refer to Paragraph C.2.6, Perform Redo Log Backup.

C.1.4.3 Online Backups

An Oracle database that can be accessed by the users is operational or **online.** Oracle database backups that occur while the database is online are called **online backups**. GCCS full and cumulative backups are online backups.

Full backups create backup versions of the data files for all tablespaces in the database. Cumulative backups create backup versions of the data files for each tablespace that has been modified since the last full backup. A tablespace is considered modified if any of its data files have changed. Once these backup data files are created, they are moved to tape.

If various changes are being made to the database while an online backup is in progress, one or more of the resulting backup data files will be **inconsistent**. Consider the following example to understand how inconsistent backup files are created during a full or cumulative backup, and then used along with redo logs (online and archived) during database recovery.

Refer to Figure C-2. A backup version of an example data file **data3.dbf** is being created. **data3.dbf** is made up of four data blocks. The data in each block is represented by a letter.

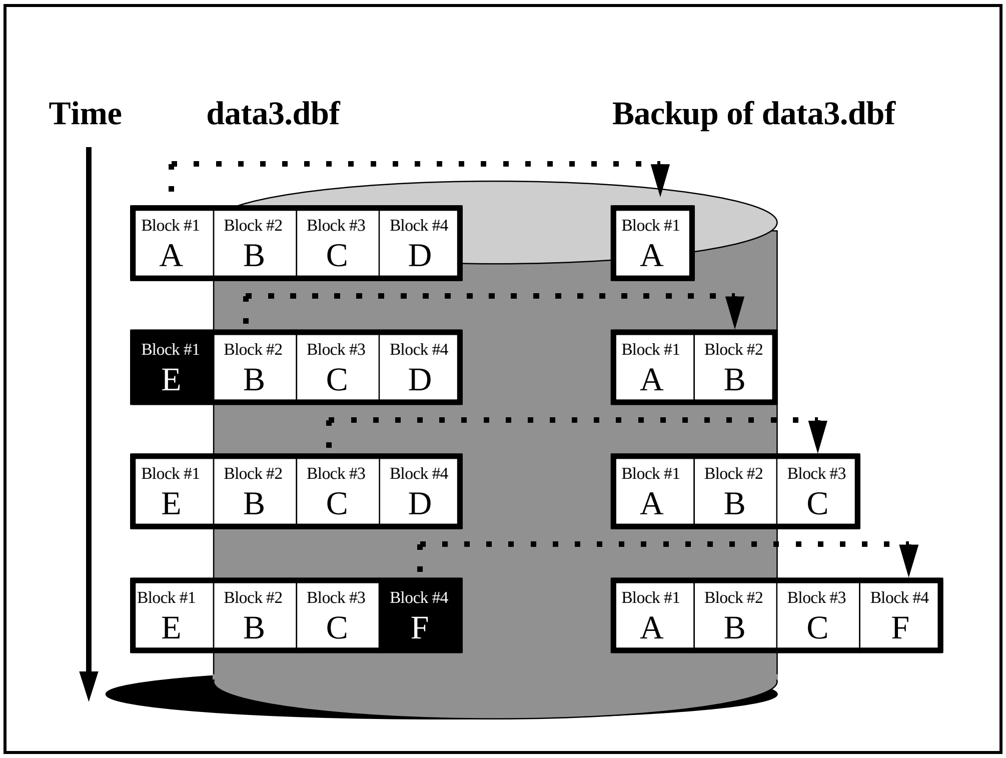


Figure C-2: An Example of an Online Database Data File Backup.

The following actions take place with respect to time.

- At the first instant in time, Block #1 of **data3.dbf** is written to the backup file.

- At the second instant in time, Block #2 of **data3.dbf** is written to the backup file. At the same time, user activity causes the system to change the data in Block #1 from A to E. Notice that the change to Block #1 is not reflected in the backup file.

- At the third instant in time, Block #3 of **data3.dbf** is written to the backup file.

- At the fourth instant in time, user activity causes the system to change the data in Block #4 of **data3.dbf** from D to F. This modified block is written to the backup file.

At this point you can see why redo log files are needed. Block #1 and Block #4 were changed, but only the change in Block #4 was captured by the backup file.

All changes to the database are captured in the redo log files. Let us say for the purposes of our example, that both changes generated in Figure C-2 are captured in the same redo log file (Figure C-3).

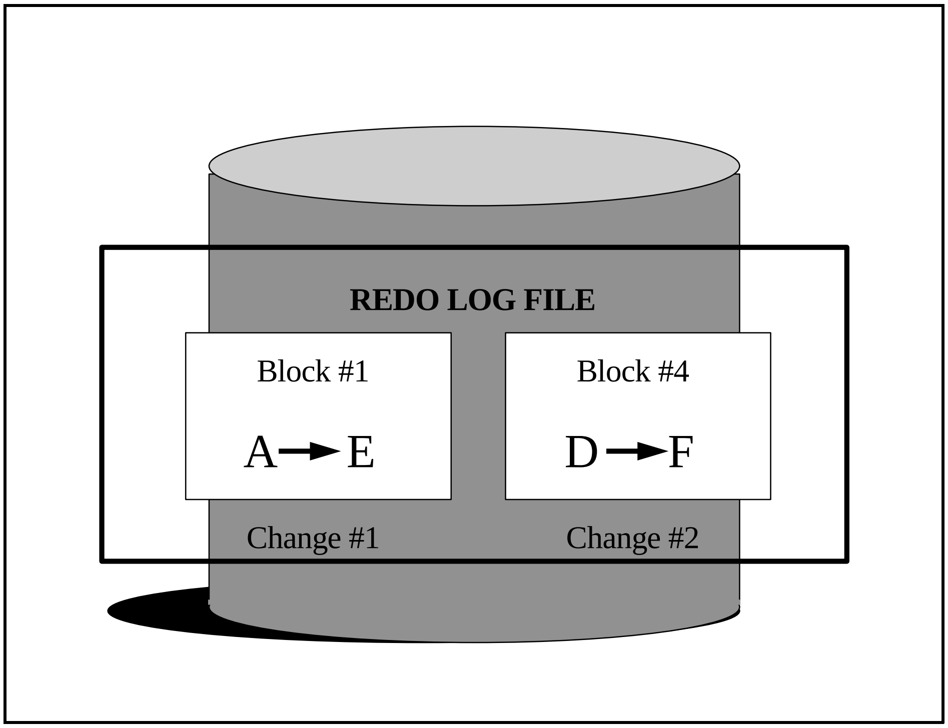


Figure C-3: Changes Recorded During the Example Online Database Data File Backup

Imagine that due to some type of failure the entire database must be recovered. When the full restore program is executed the backup version of **data3.dbf** will be copied from tape to disk, effectively becoming the new **data3.dbf**. As we saw in Figure C-2 the new **data3.dbf** is not complete. When we execute the restore redo log program, the following actions take place to make the new **data3.dbf** consistent.

- Block#1 will be updated from A to E.
- Block#4 will be verified to contain the correct information (F).

All redo logs should be preserved, that were generated since the full backup. Even after a cumulative backup has been restored, the database may still need some redo logs that were generated before the cumulative backup took place. Although the database may be recovered by restoring only a full backup and then applying redo logs. In some cases restoring the cumulative backup can save time(particularly in the case of heavy database update activity). If you choose to restore the cumulative backup during recovery, first restore from the full backup; next restore from the cumulative backup; then restore from the redo log backup. The cumulative backup is also useful for advanced recovery techniques to restore only an isolated tablespace. For more information on Backup and Recovery restore procedures refer to Section C.4, Data Restoration Procedures.

C.1.4 BACKUP AND RECOVERY APPROACH

Backup and recovery includes both automated and manual procedures (Figure C-4). Database backup procedures provide the capability to perform periodic saves of the database and archived redo logs to tape to allow for recovery of database structures and data. Recovery procedures provide the capability to restore the database back to normal after a hardware, software, network, process, or system failure.

Backup and recovery of the GCCS Database uses automated procedures wherever possible to free the site administrator from unnecessary work. Even with extensive automation, technical knowledge of Sun Unix and ORACLE7 is required to assure recovery from all failure conditions. Refer to Section C.6, Administrator Qualification, for more information on Unix and ORACLE qualifications.

The objectives for backup and recovery are to minimize data loss and minimize recovery time. This is accomplished by maintaining periodic backups and a journal of database changes (redo logs) between backups. The primary recovery mechanism is operating system disk-mirroring; backup and recovery software provides a second level of recovery assurance, as well as a recovery mechanism where both copies of a mirrored disk file are lost. The database will remain online, committing changes, during each backup.

The GCCS Database backup approach provides fail-safes to ensure database recovery. Table C-1 illustrates the primary recovery assurance measures.

Table C-1: GCCS C/S Database Recovery Assurance Measures.

RECOVERY ASSURANCE MEASURES	DESCRIPTION
Mirror control files	Mirror control files on multiple disks to eliminate a single point of failure. (Each control file is interchangable.) A duplicate control file can be copied in the event of media failure.
Safeguard archived redo logs	Isolate archived redo logs on an independent disk.
Isolate tape storage for each backup operation	Store backup files from each backup operation on different tape volumes to guard against user error or media damage.

With these recovery assurance measures in place, most recovery operations may be routinely performed by trained administrators.

C.1.6 BACKUP PROCEDURES

Backup procedures permit the administrator to make tape backups of the entire database (full backup) and backups of only tablespaces that have changed (cumulative backup). In addition, backup procedures move the archived redo logs, used to recover the database, to tape (redo log backup).

The automatic backup feature allows the backup schedule to be tailored for each site. All automated procedures may also be accessed manually. Automatic backups are initially disabled; the administrator may perform manual backups or establish an automatic schedule. Archived redo log backups, when enabled, automatically execute when the backup disk exceeds 50% capacity and the backup tape drive is free.

The C/S backup procedure includes utilities to change the automatic backup schedule and to disable automatic backups. Additional utilities allow reporting of the current backup status and termination of the backup in progress.

Each site should follow consistent procedures for tape library management and updates to the Backup Binder. The basis for development of these procedures is provided in Section C.3, Backup Library Management.

To get started quickly (recommended for experienced administrators only) refer to Paragraph C.2.1, Backup and Recovery Menu: Backup Options.

C.1.7 RECOVERY PROCEDURES

Recovery procedures consist of restore procedures and ORACLE/Unix commands performed on a case-by-case basis. ORACLE and Unix documentation (refer to Paragraph C.1.2, Referenced Documents) describes diagnosis of recovery conditions and ORACLE/Unix recovery commands. Paragraph C.5.2, Media Failure, identifies common media failure conditions with suggested ORACLE commands and restore procedures.

Restore procedures are used to move backups from tape to disk, and complement ORACLE's step by step recovery procedures. Restore procedures must be initiated manually. These procedures are most frequently used to recover damaged or corrupted media.

The three restore procedures are:

- Restore full backup
- Restore cumulative backup
- Restore redo log backups.

The restore procedures described in this document should be used to complement database recovery commands.

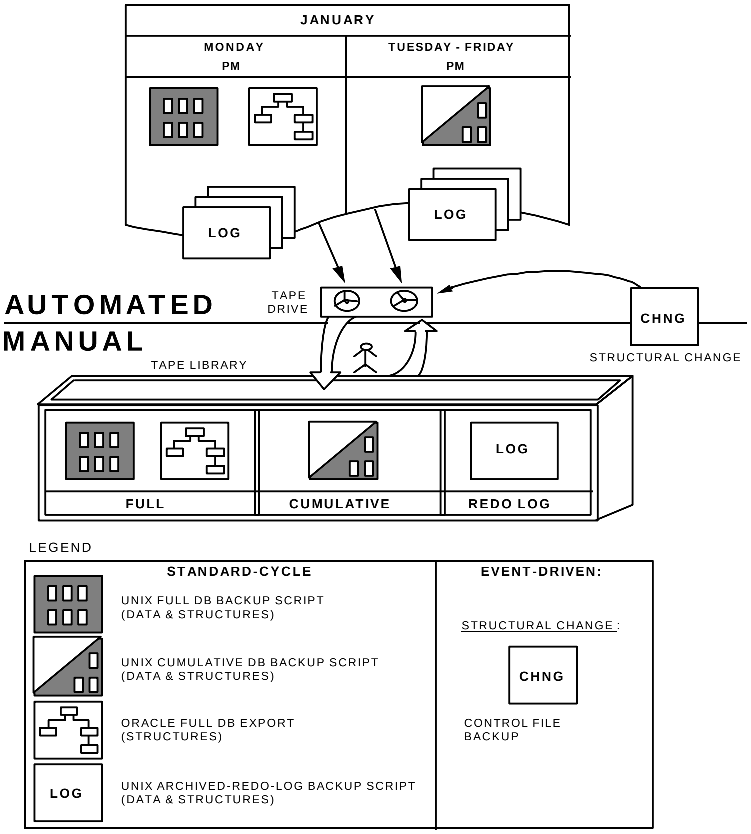


Figure C-4: GCCS C/S Standard Automated Online Backup Schedule.

C.1.8 DEVICE/DATABASE SETUP PROCEDURES

Device/database setup procedures allow you to modify the database name, tape device, and printer destination environment variables used in all Backup and Recovery programs. Refer to Paragraph C.2.2, Device/Database Settings, to view and print current environment variable settings. Before modifying environment variable settings, review Section C.2, Backup Procedures, and Section C.4, Data Restoration Procedures.

All backup and restore programs use the database name identifier stored in the **ORACLE_SID** setup file located in the **$ORACLE_HOME/RECOVERY/etc/** directory path. It is recommended that the default SID entry "GCCS" not be modified without prior coordination with the Database expert site or Backup and Recovery software developers.

Full, cumulative, and redo log backup and recovery programs set tape device environment variables to the contents of the **FULL_TAPE**, **CUM_TAPE**, **REDO_TAPE** and **DEFAULT_TAPE** setup files. These files are located in the **$ORACLE_HOME/RECOVERY/etc/** directory path. The **DEFAULT_TAPE** setup file is required for program operation and must contain the name of a tape device. The **FULL_TAPE**, **CUM_TAPE** and **REDO_TAPE** setup files are optional and do not require a tape device entry. The **DEFAULT_TAPE** setup file is used to define the tape device for any Backup or Restore program which does not have an entry in the optional setup file. The optional setup files can be used to define tape devices for the full, cumulative or redo log backup and restore programs.

Note: If the site tape device supports high density backups, it can achieve drastically faster backup and restore run times and use considerably fewer tapes. The high density device is specified using /dev/rmt/0hn instead of /dev/rmt/0n or /dev/rmt/0mn.

The purpose of the **FULL_TAPE**, **CUM_TAPE** and **REDO_TAPE** setup files is to dedicate specific tape drives to backup and restore programs. If your site's hardware configuration includes two tape drives dedicated to GCCS C/S Backup and Recovery, it would be advantageous to assign one drive to the Redo Log Backup program and assign the other drive as the default drive (used by all the other Backup/Restore programs). This setup would eliminate the need to switch between Redo Log Backup tapes, and Full/Cumulative Backup tapes. To specify the setup described, edit the **REDO_TAPE** file to specify the device name of the first tape drive, then edit the **DEFAULT_TAPE** file to specify the device name of the second tape drive. The tape device name must specify "no rewind" (i.e., *dev/rmt/0hn*).

Note: The process of assigning different tape drives to backup programs will not allow Backup or Restore programs to run simultaneously on the same database server. Only one program can run at a time.

The printer destination setup file is used to direct Backup and Recovery printouts to a specific printer. The Backup and Recovery programs set the *lp -d* switch to the contents of the **PRINTER** setup file located in the **$ORACLE_HOME/RECOVERY/etc/** directory path. The **PRINTER** setup file is optional and does not require a printer destination entry. If an entry is omitted, the default printer is used to print out the reports. To define a different *lp* destination, edit the **PRINTER** file to specify the name of the new default printer. A current list of installed printers can be obtained by issuing the <*lpstat -t*> command at the Unix prompt.

An invalid entry in any of the required or optional tape device/printer setup files will result in *mt*, *tar* and/or *lp* command errors, during execution of Backup and Recovery programs.

SECTION C.2 — BACKUP PROCEDURES

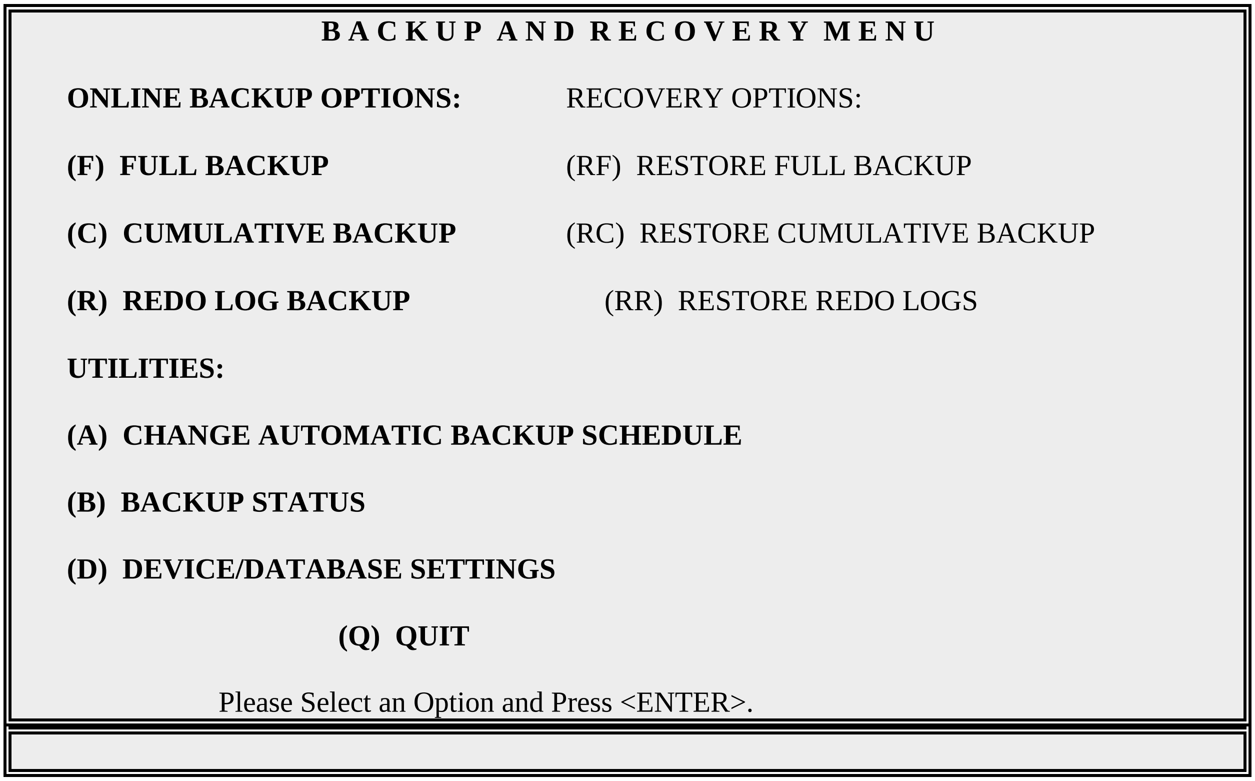
C.2.1 BACKUP AND RECOVERY MENU: BACKUP OPTIONS

Backup procedures include full backups, cumulative backups, redo log backups, and structural change backups. Backup procedures also cover changes to the automatic backup schedule and backup status (including termination of the backup in progress).

Full, cumulative, and redo log backups can be executed automatically, or selected manually from the Backup and Recovery menu. Changes to the automatic backup schedule, and backup status checks must be selected manually from the BACKUP AND RECOVERY MENU. Structural change backups should be performed using ORACLE commands; this straightforward operation is not included in the BACKUP AND RECOVERY MENU.

Accessing backup and recovery software - To access backup and recovery software, log onto the database server directly from the GCCS "globe" as the unix user <*oradba*> (do not *su* from another account). By default, you should be in the home directory of the <*oradba*> Unix account. Enter <*br_main*> to pull up the BACKUP AND RECOVERY MENU screen.

Figure C-5 shows the backup options available from the BACKUP AND RECOVERY MENU.



BACKUP AND RECOVERY MENU

ONLINE BACKUP OPTIONS:	RECOVERY OPTIONS:
(F) FULL BACKUP	(RF) RESTORE FULL BACKUP
(C) CUMULATIVE BACKUP	(RC) RESTORE CUMULATIVE BACKUP
(R) REDO LOG BACKUP	(RR) RESTORE REDO LOGS

UTILITIES:

(A) CHANGE AUTOMATIC BACKUP SCHEDULE

(B) BACKUP STATUS

(D) DEVICE/DATABASE SETTINGS

(Q) QUIT

Please Select an Option and Press <ENTER>.

Figure C-5: Backup and Recovery Menu — Backup Options.

To leave the BACKUP AND RECOVERY MENU, enter (Q) QUIT. You will return to the default $ORACLE_HOME directory.

Automatic backups are initially disabled. This allows you to gain familiarity with the system and determine site requirements before custom-tailoring an automatic backup schedule. A typical schedule for automatic backups, referred to as the standard schedule, provides a baseline schedule for backup planning.

Note: It is critical that you set the automatic archived redo log backup schedule (refer to Paragraph C.2.3.3, Change Automatic Backup Schedule) each time backup and recovery software is installed. Failing to correctly enable redo log backups may cause the directory /oracle/smback/arch to fill to capacity and halt the database.

Backup procedures and initial settings are illustrated in Table C-2. The standard schedule is provided to assist with backup planning.

Table C-2: Backup Procedures.

PROCEDURE	BACKUP & RCVY MENU	AUTOMATIC BACKUP	INITIAL SETTING	STANDARD SCHEDULE	PARAGRAPH
Device/Database setup procedures (modify)	NO	-	-	-	C.1.7
Device/Database settings (view)	YES	-	DEFAULT	-	C.2.2
Change auto backup schedule	YES	-	-	-	C.2.3.3
Full backup	YES	YES	DISABLED	1 per week	C.2.4
Cumulative backup	YES	YES	DISABLED	1 per day	C.2.5
Redo log backup	YES	YES	DISABLED (Disregard backup times shown on screen)	At 50% capacity; check every 15 minutes	C.2.6
Backup status	YES	-	-	-	C.2.7
Terminate backup	YES	-	-	-	C.2.8
Structural change backup	NO	NO	-	-	C.2.9

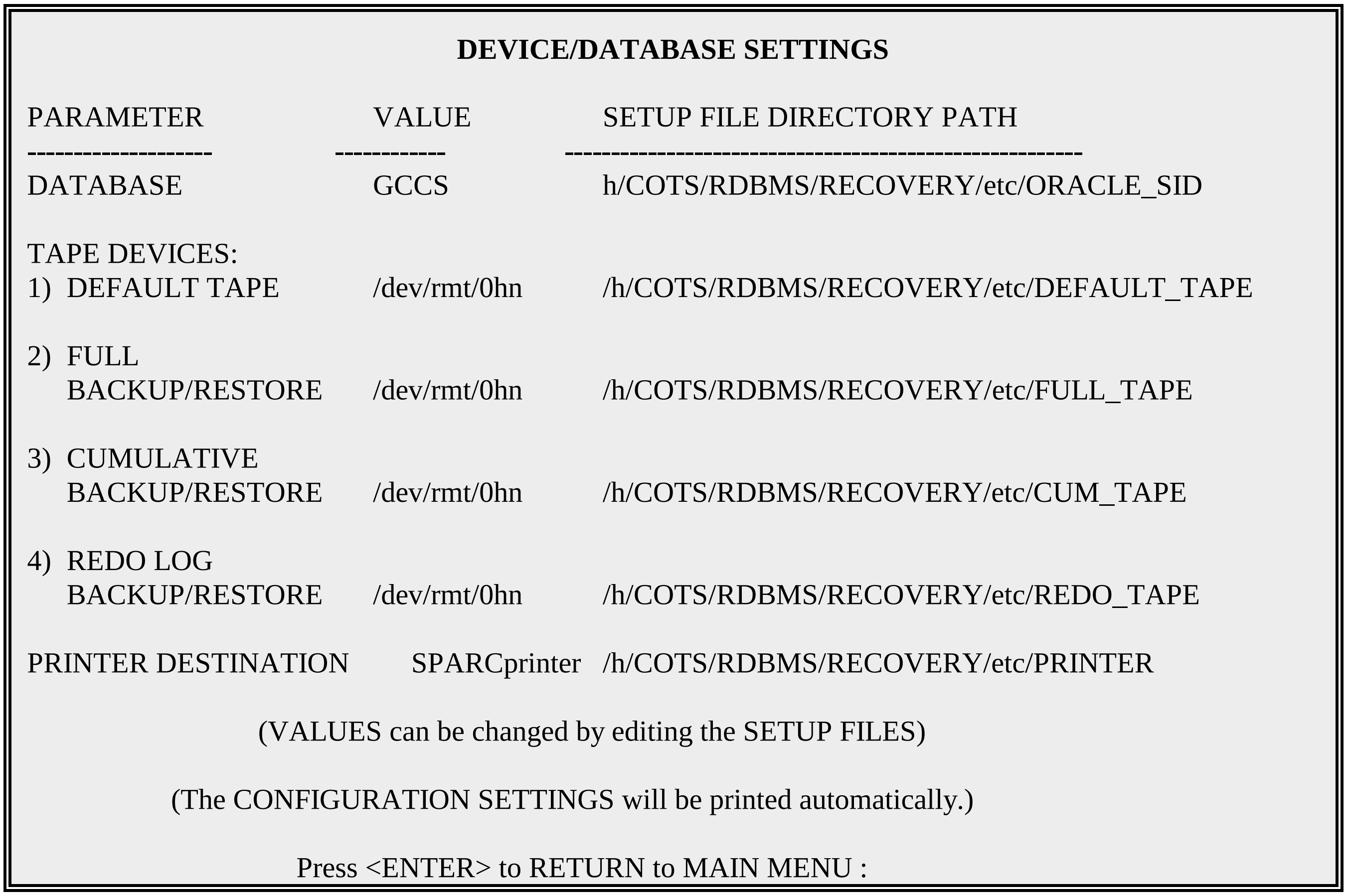
C.2.2 DEVICE/DATABASE SETTINGS

The DEVICE/DATABASE SETTINGS screen (Figure C-6) displays and prints default variable settings for the database name, default tape drive, and default printer destination. To view this screen, select (D) DEVICE/DATABASE SETTINGS from the Backup & Recovery menu (Figure C-6). To modify the database name, tape device, or printer destination environment variables, refer to Paragraph 1.8, Device/Database Setup Procedures.

The DEVICE/DATABASE SETTINGS screen provides the values and directory paths of all environment variables used by the backup/restore programs. Environment variables include the database name, the tape device set for each type of backup/restore, and the printer assignment.

Each environment variable is named in the PARAMETER column. The current setting for each parameter is shown in the center column, labelled VALUE. The location of the file containing the parameter setting is shown in the far right column, labelled SETUP FILE DIRECTORY PATH. To modify any parameter setting, refer to Paragraph 1.8, Device/Database Setup Procedures.

Press ENTER to return to the BACKUP AND RECOVERY MENU. The device/database settings will automatically print to the default printer after exiting this screen.



```
                         DEVICE/DATABASE SETTINGS

PARAMETER              VALUE          SETUP FILE DIRECTORY PATH
--------------------   ------------   -----------------------------------------------------
DATABASE               GCCS           h/COTS/RDBMS/RECOVERY/etc/ORACLE_SID

TAPE DEVICES:
1)  DEFAULT TAPE       /dev/rmt/0hn   /h/COTS/RDBMS/RECOVERY/etc/DEFAULT_TAPE

2)  FULL
    BACKUP/RESTORE     /dev/rmt/0hn   /h/COTS/RDBMS/RECOVERY/etc/FULL_TAPE

3)  CUMULATIVE
    BACKUP/RESTORE     /dev/rmt/0hn   /h/COTS/RDBMS/RECOVERY/etc/CUM_TAPE

4)  REDO LOG
    BACKUP/RESTORE     /dev/rmt/0hn   /h/COTS/RDBMS/RECOVERY/etc/REDO_TAPE

PRINTER DESTINATION       SPARCprinter   /h/COTS/RDBMS/RECOVERY/etc/PRINTER

              (VALUES can be changed by editing the SETUP FILES)

         (The CONFIGURATION SETTINGS will be printed automatically.)

                 Press <ENTER> to RETURN to MAIN MENU :
```

Figure C-6: Device/Database Settings Screen.

C.2.3 AUTOMATIC BACKUPS

Automatic backups consist of time-initiated and capacity-initiated programs. The full and cumulative backups are time-initiated, and the archived redo log backup is capacity-initiated. The full and cumulative backups copy data from disk to tape. The archived redo log backup removes data from the backup disk and writes it to tape, to free disk space. The purpose of the redo log backup is to free disk space in the directory /oracle/smback/arch.

C.2.3.1 Automatic Backup Schedule

Automatic backups are initiated on a designated time schedule. Each type of backup may be enabled or disabled.

Initially, all automatic backups are disabled (disregard any setting indicating that backups are enabled). The standard automatic backup schedule is depicted in Table C-3. This schedule is intended for a single-shift, Monday through Friday site. It may be used as a starting point to tailor individual site schedules.

Table C-3: Standard Automatic Backup Schedule.

AT 1800 HRS ON:	**SUN**	**MON**	**TUE**	**WED**	**THU**	**FRI**	**SAT**
FULL BACKUP		X					
CUM BACKUP			X	X	X	X	
REDO LOG BACKUP	Check for > 50% capacity at 10, 25, 40, 55 minutes after the hour						

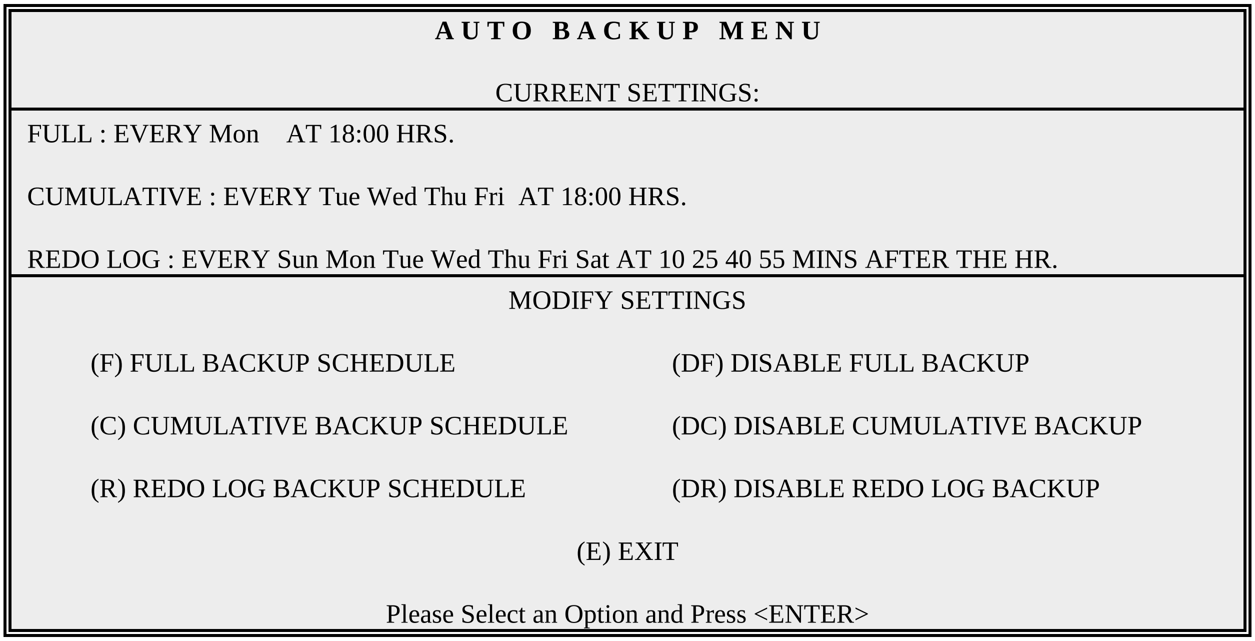
All backup scripts (including the redo log backup) are initially triggered at a time set in the Unix **crontab** file. Backups execute based on system time as set for the Unix user <*oradba*>; changes made to the time setting for the Unix user <*oradba*> WILL affect the real-time backup schedule. To determine the current time for the user <*oradba*>, log in from the globe as unix user <*oradba*> and enter <*date*>.

Archived redo log backups differ from the other backups because the backup process is capacity-initiated based on the results of periodic capacity checks. Archived redo log backups are initiated when the backup filesystem (/oracle/smback) exceeds 50% capacity. The Unix **crontab** file triggers a Unix script to check free space on the backup filesystem at times specified in the automatic backup schedule (standard schedule is every 15 minutes). The script checks free space and determines if a redo log backup is required. A redo log tape must be in the drive, and no other backup/restore procedures may be in progress for the automatic redo log backup to execute.

Full, cumulative, and archived redo log backups may also be initiated manually through the BACKUP AND RECOVERY MENU.

C.2.3.2 View Automatic Backup Schedule

To view the automatic backup schedule, select the option (A) CHANGE AUTOMATIC BACKUP SCHEDULE, from the BACKUP AND RECOVERY MENU. The AUTO BACKUP MENU, shown in Figure C-7, will appear. The current **crontab** file settings for the full backup, cumulative backup, and redo log check-times are displayed at the top of the screen. Settings are displayed by day of the week, hour and minute.



```
                              AUTO BACKUP MENU

                             CURRENT SETTINGS:
FULL : EVERY Mon    AT 18:00 HRS.

CUMULATIVE : EVERY Tue Wed Thu Fri  AT 18:00 HRS.

REDO LOG : EVERY Sun Mon Tue Wed Thu Fri Sat AT 10 25 40 55 MINS AFTER THE HR.
                              MODIFY SETTINGS

      (F) FULL BACKUP SCHEDULE                 (DF) DISABLE FULL BACKUP

      (C) CUMULATIVE BACKUP SCHEDULE           (DC) DISABLE CUMULATIVE BACKUP

      (R) REDO LOG BACKUP SCHEDULE             (DR) DISABLE REDO LOG BACKUP

                                  (E) EXIT

                   Please Select an Option and Press <ENTER>
```

Figure C-7: Auto Backup Menu.

Any or all automated backups may be disabled from this menu. If there are no **crontab** file entries for a type of backup, **"DISABLED"** appears after the type of backup.

Note: If backups are disabled and manual backups are not performed, local recovery from a database failure may be impossible. If Redo Log backups are disabled, the redo log storage space may fill to capacity and halt the database. Always check the redo log backup settings after backup and recovery software is installed.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.3.3 Change Automatic Backup Schedule

One of the most powerful features of the backup approach is the ability to customize the automatic backup schedule to meet site demands. Such a change may be necessary to accommodate varying levels of activity at different sites, or increases or decreases in activity at a particular site.

For example, during an exercise or crisis, when system use may increase dramatically, the backup schedule could be modified to perform full backups daily instead of weekly, and to disable cumulative backups.

Changes to the automatic backup schedule should only be made by an experienced administrator who understands all aspects of the backup approach. Administrative procedures, including maintenance of the Backup Binder and Tape Library, must be updated to reflect changes to the backup schedule.

Error conditions may occur while tailoring the automatic backup schedule to site requirements. Common error conditions, and troubleshooting actions are shown in Table C-4.

Table C-4: Troubleshooting Auto Backups.

ERROR CONDITION	ACTION
The word **"error"** appears in a CURRENT SETTINGS line on the AUTO BACKUP MENU.	1. An invalid entry exists in the **crontab** file for this type of backup. MODIFY each schedule displaying an **"error"** time. Invalid entries will be cleared, and the times just entered will appear as current settings. 2. Invalid **crontab** file entries include: • Day or hour times specified for redo log backup check times • Range of time specified (i.e., 10-15) • Wildcard time specified (i.e., *).
The message **"AUTO BACKUP SETTINGS NOT FOUND"** appears on each CURRENT SETTINGS line on the AUTO BACKUP MENU.	1. If a full, cumulative, or redo log restore is in progress, DO NOT modify the auto backup settings. The restore program temporarily disables the auto backup facility. 2. If no restore is in progress, MODIFY each schedule line. Invalid entries will be cleared, and the times just entered will appear as current settings.
An error message appears after you enter data to MODIFY a backup schedule.	1. Check your entries carefully against the examples on screen. Are your entries unique, sequential integers? 2. If you continue to have problems after correcting syntax errors, there may be other syntax problems in the table. Obtain a printout and contact the network TDBM.

ERROR CONDITION	ACTION
Full or cumulative backups do not execute automatically.	1. Check current settings on the auto backup menu. If **"Disabled"** appears for full or cumulative backup, MODIFY full or cumulative backup to specify a backup time. 2. Check current settings on the AUTO BACKUP MENU. No two backups should be set to execute at the same day and time. 3. Check the full and cumulative backup status under the STATUS MENU. If **"TERMINATED"** appears, the error message will indicate possible problems.

C.2.3.3.1 Set Automatic Backup Times. To set automatic backup times, first select (A) CHANGE AUTOMATIC BACKUP SCHEDULE from the BACKUP AND RECOVERY MENU. The AUTO BACKUP MENU will appear. To modify backup times or set new times for a disabled backup, select (F) FULL BACKUP SCHEDULE or (C) CUMULATIVE BACKUP SCHEDULE. To set redo log check time select (R) REDO LOG BACKUP SCHEDULE.

Full and cumulative backups may be set to initiate at any time 24 hours per day, 7 days per week. Redo log backups may be set to initiate at 0-59 minutes after each hour. Because only one backup operation will run at the same time, automatic backup times should be scheduled to minimize conflicts.

Follow syntax instructions on the data-entry screen carefully. No duplicate numbers should be entered. If your changes are committed successfully, the Current Settings screen will be updated.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.3.3.2 Disable Automatic Backups. To disable an automatic backup, first select (A) CHANGE AUTOMATIC BACKUP SCHEDULE, from the BACKUP AND RECOVERY MENU. The AUTO BACKUP MENU will appear. Select (DF) DISABLE FULL BACKUP, (DC) DISABLE CUMULATIVE BACKUP or (DR) DISABLE REDO LOG BACKUP.

Disabling redo log backup check-times freezes the **crontab** file script that periodically checks backup disk capacity. It does not disable ORACLE redo log archiving. If redo log backups are disabled, the redo log storage space may fill to capacity and stop the database.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.4 PERFORM FULL BACKUP

The full backup is initiated either automatically as set in the AUTO BACKUP MENU, or manually from the BACKUP AND RECOVERY MENU. To initiate the full backup automatically, specify the desired time and day using the AUTO BACKUP MENU (refer to Paragraph C.2.3.3). To initiate the full backup manually, select (F) FULL BACKUP from the BACKUP AND RECOVERY MENU. Ensure that the full backup tape is inserted in the tape drive.

Because the full backup runs in background mode, you may use the terminal for other tasks while the procedure is running. Errors encountered are mailed to the <*oradba*> user account, and on-screen prompts are sent to the <*oradba*> and <*secman*> user accounts. You should open a session to one of these accounts to receive backup message prompts whenever the full backup program is run. In addition, you may check the progress of the backup and any errors encountered so far, through the BACKUP STATUS MENU.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.4.1 Full Backup Execution

The full backup procedure runs in background, and backs up all tablespaces in the GCCS ORACLE database. The database remains online, and modifications may be made while the backup is executing.

The program determines the tablespace names and associated files from the **DBA_DATA_FILES** table. After the tablespace files are generated, the program copies them to tape serially, one tablespace at a time. Each tablespace is stored in a tar file of the same name on the backup tape. After all tablespaces are written to tape, the control file and init.ora file are copied to tape. Finally, an export of the database structure is performed. Because database imports are considerably slower than the tar command, full database exports are not used to backup large volumes of data.

Only one backup or restore operation should ever run at a time. Before execution, the full backup program checks for any other backup or restore processes currently running, and establishes a lock if none are found. This prevents errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the full backup program. Errors may be classified as warning or terminating. Warning errors generally occur after all tablespaces have been moved to tape and include errors encountered during control file backup and **V$FILESTAT** (used for cumulative backup) updates. Most errors are terminating errors and include write and permission errors, tar I/O errors, locking errors (another backup or restore is currently running), and tape timeout errors (program cannot find the correct tape after approximately three hours). All errors are written to three files: the **error.log** (accumulation of all error messages), **current_progress** (describes all program events), and **exit_status** (backup summary log filed in Backup Binder). After errors are written to these files, the error message is sent to the <*oradba*> user's mail account.

In the event of program termination you must correct the problem and manually initiate the full backup again from the Backup and Recovery menu. Warning errors are less critical errors that should be investigated, but generally do not require restarting the full backup manually.

When inserting a tape, the software confirms that either the tape is blank or new, or that the correct header is on the tape. The header is actually the filename of the first file placed on the tape using the Unix **tar** command. For the Full Backup, the filename <*FULL_x*> is sought as the first file on tape, where x is a sequence number beginning with 1. This sequence number in the header enables the software to handle a multiple tape backup. The software waits approximately three hours before termination in the event an unexpected backup tape is in the drive at the beginning of a backup, or if a subsequent tape needs to be inserted for a multiple tape backup.

C.2.4.2 Full Backup Completion

After execution of the full backup, a backup summary log is printed, providing complete backup reference information. This log contains the type of backup, the date and time, and the final status of the program (SUCCESSFUL, SUCCESSFUL with WARNINGS, or TERMINATED). If the program was SUCCESSFUL, the printout will also list the files copied to tape, grouped by tablespace name. You should file this printout under Full Backups in the Backup Binder; it will be an important reference if recovery is required.

C.2.5 PERFORM CUMULATIVE BACKUP

The cumulative backup is initiated either automatically as set in the AUTO BACKUP MENU, or manually from the BACKUP AND RECOVERY MENU. To initiate the cumulative backup automatically, specify the desired time and day using the AUTO BACKUP MENU (refer to Paragraph C.2.3.3). To initiate the cumulative backup manually, select © CUMULATIVE BACKUP from the BACKUP AND RECOVERY MENU. Ensure that the cumulative backup tape is inserted in the tape drive.

Because the cumulative backup runs in background mode, you may use the terminal for other tasks while the procedure is running. Any errors encountered are logged in the exit_status file and the current_progress file. Errors are mailed to the <*oradba*> user account, and on-screen prompts are sent to the <*oradba*> and <*secman*> user accounts. You should open a session to one of these accounts to receive backup message prompts whenever the cumulative backup program is run. In addition, you may check the progress of the cumulative backup and any errors encountered so far, through the STATUS MENU.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.5.1 Cumulative Backup Execution

The cumulative backup procedure runs in background, and backs up all tablespaces in the ORACLE database that have changed since the last full backup. The database remains online, and modifications may be made while the backup is executing. The program determines all files that have changed by comparing blocks written at the last full backup to blocks written now (**V$FILESTAT**). All tablespaces containing even one changed file are added to the cumulative backup list. If tablespace Input/Output (I/O) activity cannot be determined (i.e., can't access V$FILESTAT table, etc.), a full backup is performed. After the tablespace files are generated using the **DBA_DATA_FILES** table, the program copies them to tape serially, one tablespace at a time. Each tablespace is stored in a tar file of the same name on the backup tape. After all tablespaces are written to tape, the control file is copied to tape.

Only one backup or restore operation should ever run at one time. Before execution, the cumulative backup program checks for any other backup or restore processes currently running, and establishes a lock if none are found. This prevents errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the cumulative backup program. Errors may be classified as warning or terminating. Warning errors generally occur after all tablespaces have been moved to tape and include errors encountered during control file backup. Most errors are terminating errors and include write and permission errors, tar I/O errors, locking errors (another backup or restore is currently running), and tape timeout errors (program cannot find the correct tape after approximately three hours). All errors are written to three files: the **error.log** (accumulation of all error messages), **current_progress** (describes all program events), and **exit_status** (backup summary log filed in Backup Binder). After errors are written to these files, the error message is sent to the <*oradba*> user's mail account.

In the event of program termination you must correct the problem and manually initiate the cumulative backup again from the BACKUP AND RECOVERY MENU. Warning errors are less critical errors that should be investigated, but generally do not require restarting the cumulative backup manually.

When inserting a tape, the software confirms that either the tape is blank or new, or that the correct header is on the tape. The header is actually the filename of the first file placed on the tape using the Unix **tar** command. For the Cumulative Backup, the filename <*CUM_x*> is sought as the first file on tape, where x is a sequence number beginning with 1. This sequence number in the header enables the software to handle a multiple tape backup. The software waits approximately three hours before termination in the event an unexpected backup tape is in the drive at the beginning of a backup, or if a subsequent tape needs to be inserted for a multiple tape backup.

C.2.5.2 Cumulative Backup Completion

After execution of the cumulative backup, a backup summary log is printed, providing complete backup reference information. This log contains the type of backup, the date and time, and the final status of the program (SUCCESSFUL or TERMINATED). If the program was SUCCESSFUL, the printout will also list the files copied to tape, grouped by tablespace name. You should file this printout under Cumulative Backups in the Backup Binder; it will be an important reference if recovery is required.

C.2.6 PERFORM REDO LOG BACKUP

The archived redo log backup is initiated either automatically (capacity-initiated, based on backup disk free space), or manually from the BACKUP AND RECOVERY MENU. The space check procedure executes automatically at the times shown in the AUTO BACKUP MENU. If the backup filesystem is above 50% capacity, the redo log backup is initiated. The backup filesystem is the filesystem which contains the directory to which Oracle writes redo log files (i.e., /oracle/smback).

The software also supports an emergency procedure to write redo logs in a controlled manner, to free space on all filesystems owned by <*oracle*> (also referred to as contingency filesystems). This feature is included in the software, but will not execute at the present time, because no contingency filesystems are owned by <*oracle*>. References to the contingency redo log backup capability have been maintained in this document.

To initiate the redo log backup manually, select (R) REDO LOG BACKUP from the BACKUP AND RECOVERY MENU. Ensure that the redo log backup tape is inserted in the tape drive.

Because the archived redo log backup runs in background mode, you may use the terminal for other tasks while the procedure is running. Errors are mailed to the <*oradba*> user account, and on-screen prompts are sent to the <*oradba*> and <*secman*> user accounts. You should open a session to one of these accounts to receive backup message prompts whenever the redo log backup program is run. In addition, you may check the progress of the backup and any errors encountered so far, through the STATUS MENU.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.6.1 Redo Log Backup Execution

The archived redo log procedure runs in background, and backs up all redo logs present on the backup disk from the oldest log number to the most recent log number. After each redo log is successfully copied to tape, it is deleted from the backup disk. The database remains online, and modifications may be made while the backup is executing. Each redo log is moved to a tar file of the same name on the archived redo log backup tape. The archived redo log procedure maintains a sequence count to ensure that no redo log numbers are skipped.

Note: The contingency redo log backup procedures described in this paragraph are for general reference only. Under the current system configuration, the contingency redo log backup function will not execute (refer to Paragraph C.2.6, Perform Redo Log Backup).

If the backup disk reaches a set maximum capacity, redo logs are automatically diverted to free space on any other filesystem owned by ORACLE. This allows ORACLE to continue writing redo logs to these contingency drives, and thus continue processing. Warning messages will be sent to the oracle user mail account and error logs: **"The archive space usage is at xx%. The excess redo log files will be relocated."** You should take action to copy redo logs to disk as soon as possible;

ORACLE will stop processing when redo log space is full. When the redo log backup procedure is initiated again, redo logs are automatically retrieved from contingency locations and moved to tape in sequential order.

Only one backup or restore operation should ever run at one time. Before execution, the archived redo log backup program checks for any other backup or restore processes currently running, and establishes a lock if none are found. These procedures prevent errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the archived redo log program. All archived redo log errors are terminating. Terminating errors include write and permission errors, tar I/O errors, locking errors (another backup or restore is currently running), and tape timeout errors (program cannot find the correct tape after approximately three hours). The program will also terminate if the database was offline and could not be mounted, when the program attempted to obtain the current log sequence number. (This may occur if the database is corrupted or the control file is missing). All errors are written to three files: the **error.log** (accumulation of all error messages), **current_progress** (describes all program events), and **exit_status** (backup summary log filed in Backup Binder). After errors are written to these files, the error message is sent to the <*oradba*> user's mail account with the exception of the locking error. Locking errors are written to the appropriate files but they are neither mailed nor printed.

In the event of program termination you must correct the problem and manually initiate the redo log backup again from the BACKUP AND RECOVERY MENU.

When inserting a tape, the software confirms that either the tape is blank or new, or that the correct header is on the tape. Under normal operation, you should continue to use each redo log backup tape until it fills to capacity. (For example, you complete a Redo Log Backup on tape **REDO - 27** and remove the tape to perform a Full Backup. After the Full Backup has completed, you should re-insert the tape **REDO - 27** to continue Redo Log Backups.) To prevent inadvertent data loss, you may only overwrite an earlier redo log tape if a tar input/output error occurred on the previous tape, or the previous tape was full. When overwriting an earlier redo log backup tape, ensure that the tape is fully rewound before inserting it in the drive.

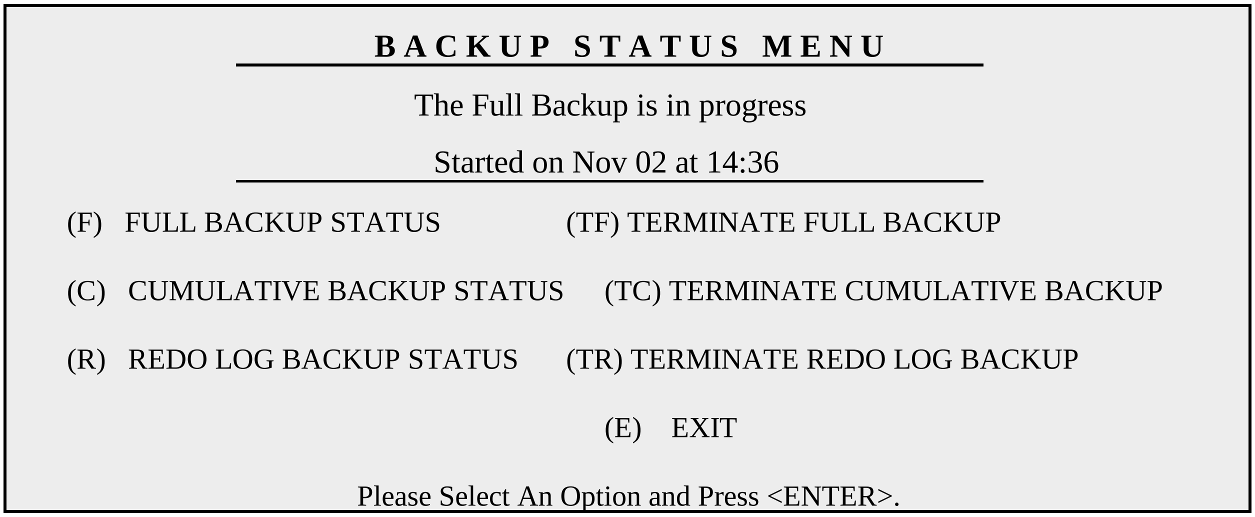
Before each Redo Log Backup, the filename <*REDO_LOG*> is sought as the first file on tape. (The software also uses the contents of this file.) This sequence number in the header enables the software to handle a multiple tape backup. The software waits approximately three hours before termination in the event an unexpected backup tape is in the drive at the beginning of a backup, or if a subsequent tape needs to be inserted for a multiple tape backup.

C.2.6.2 Redo Log Backup Completion

After execution of the archived redo log backup, a backup summary log is printed, providing complete backup reference information. This log contains the type of backup, the date and time, and the final status of the program (SUCCESSFUL or TERMINATED). If the program was SUCCESSFUL, the printout will also list the archived redo log numbers copied to tape. Each page header will show the log number of the first log copied to the physical tape. You should file this printout under Archived redo log backups in the Backup Binder; it will be an important reference if recovery is required.

C.2.7 BACKUP STATUS

The BACKUP STATUS MENU may be accessed by selecting (B) BACKUP STATUS from the BACKUP AND RECOVERY MENU. The BACKUP STATUS MENU, Figure C-8, will appear. This screen displays the name and start date/time of any backup in progress. This screen is a snapshot of the operation in progress; to update the status screen, exit and reenter the menu.



```
                    BACKUP STATUS MENU
                The Full Backup is in progress
                  Started on Nov 02 at 14:36

(F)  FULL BACKUP STATUS          (TF) TERMINATE FULL BACKUP

(C)  CUMULATIVE BACKUP STATUS     (TC) TERMINATE CUMULATIVE BACKUP

(R)  REDO LOG BACKUP STATUS      (TR) TERMINATE REDO LOG BACKUP

                                  (E)   EXIT

            Please Select An Option and Press <ENTER>.
```

Figure C-8: Backup Status Menu.

The BACKUP STATUS MENU provides the capability to view detailed progress and error information for any backup operation in progress. It also provides progress, error and completion status for the most recent backup of each type. To review this information, select the type of backup from the Status menu: (F) FULL BACKUP STATUS, (C) CUMULATIVE BACKUP STATUS, or (R) REDO LOG BACKUP STATUS. Navigate through the text output using the text control keys for your system. The completion status (SUCCESSFUL, TERMINATED WITH WARNINGS, or TERMINATED) is located in the bottom text of each entry. (Any backup currently in progress will not have a status message at the bottom of text until the operation is complete). Like the current status display, the detailed status report represents a snapshot; to update the detailed status, exit and reselect the type of backup.

The BACKUP STATUS MENU should be checked from time to time to confirm proper execution of backup operations.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.8 TERMINATE BACKUP

The BACKUP STATUS MENU provides the capability to terminate the full, cumulative or redo log backup in progress. To terminate a backup program, first access the Backup Status Menu by selecting (B) BACKUP STATUS from the BACKUP AND RECOVERY MENU. Enter (TF) TERMINATE THE FULL BACKUP, (TC) TERMINATE THE CUMULATIVE BACKUP, or (TR) TERMINATE THE REDO LOG BACKUP. A confirmation screen will display the terminate option chosen; press (C) CONTINUE WITH TERMINATION OF THE BACKUP or (Q) QUIT to return to the BACKUP STATUS MENU without terminating. The (C) option will halt execution of the backup operation and stop the tape.

Immediately after issuing the terminate command, open the tape drive and remove the tape. Reinsert the tape and close the drive. The tape will automatically position to the beginning of the tape for full or cumulative backup tapes, or the end of the last complete redo log (during the next redo log backup). If a redo log was only partially written to the old tape when the program was terminated, it will automatically write to tape during the next redo log backup.

Under some conditions, the tape drive may not respond to a terminate command. If the read/write light continues to flash, the tape drive is not responding. In this situation, the procedures remain the same; open the drive, remove the tape and then reinsert the tape.

Enter (E) EXIT to return to the BACKUP AND RECOVERY MENU.

C.2.9 STRUCTURAL CHANGE BACKUP

Certain changes made to the database require control file backups both before and after the change. These are: 1) Create or drop a tablespace; 2) Add, or rename (relocate) a data file in an existing tablespace; and 3) Add, rename (relocate), or drop an online redo log group or member.

If any of these alterations are planned, make a control file backup both before and after the alteration. Use the *ALTER DATABASE* command with the *BACKUP CONTROLFILE* option. It is recommended that these backups be stored on a separate tape(s) labeled **CHNG - (*1-10*)**, using the file naming convention **CF_*Date-time***. The tape and file name for each control file backup should be annotated on the most current full or cumulative backup report located in the Backup Binder.

C.2.10 SUPPLEMENTARY USER BACKUP

Full, cumulative and redo log backups allow the recovery of the entire database, including user permission tables, Oracle user accounts, roles, and grants. However, this information must be restored as part of a complete database recovery, and may not be accessed independently. Supplementary user backups allow backup and restoration of the user permission tables, Oracle user accounts, roles, grants and synonyms independently from complete database recovery. Supplementary user backups are written to disk.

To perform a supplementary user backup, login as the unix user <*oradba*> from the GCCS globe. Ensure that no users are on the system, or that no changes to users or user permissions are committed during backup execution. Supplementary backups are designed for a static database; if changes to user data occur, the backup information will be out of sync.

Enter <*cd /h/SMDB/Scripts/SM_bld_tables*> to change directories to the database scripts area. Start the supplementary user backup by entering <*bkup_user_perm.sh*>. During execution of the backup, six files are generated under the path **/h/SMDB/Scripts/SM_bld_tables**, as shown in Table C-5.

Table C-5: Supplementary User Backup Files Generated by bkup_user_perm.sh.

Generated File Name (/h/SMDB/Scripts/SM_bld_tables)	**Description**
bkup_user_perm.dmp	Export dump file containing compressed rows for the following user permissions tables: **JOPES_USER**, **USER_FUNCTION_PERMISSION**, **OPLAN_SERIES_PERMISSION**, **USER_OPLAN_PERMISSION**, **OPLAN_ACCESS**.
bkup_user_perm.log	Log file containing results of the user permissions export.
rstr_users.sql	Generated SQL script containing information to restore externally identified GCCS Oracle user accounts.
rstr_roles.sql	Generated SQL script containing information to restore GCCS Oracle roles.
rstr_grants.sql	Generated SQL script containing information to restore GCCS Oracle grants on database objects.
rstr_synonyms.sql	Generated SQL script containing information to restore GCCS Oracle synonyms.

For each backup, the current and previous backup files are maintained. Previous backup files are designated with the extension *.old.* For example, the previous backup file containing Oracle user accounts is named *rstr_users.sql.old.* For additional recovery assurance, you should periodically move the backup files to tape.

If desired, you may setup the supplementary user backups to execute automatically using the crontab utility. To do this, first login from the GCCS globe as the unix user <*oradba*>. Next, specify the text editor you will be using, by setting the environment variable EDITOR; to use *vi*, enter the command <*setenv EDITOR vi*>. Next, edit the <*oradba*> user's cron file by entering the command <*crontab -e*>. Using text editor commands, add a one line entry below all existing lines, to execute the backup job at a specified time (refer to the unix man-page "crontab" for more information on job scheduling). Take care not to alter any existing lines in the file. A sample entry to execute supplementary user backups every day at 0400 hrs would appear as follows:

```
00  04  *  *  *  /h/SMDB/Scripts/SM_bld_tables/bkup_user_perm.sh
```

When you have finished editing the crontab file, save the file using text editor commands. Review the resulting crontab file by entering the command <*crontab -l*>. Finally, clean up the environment variable setting by entering <*unsetenv EDITOR*>. Always check the automatic backup schedule through the AUTO BACKUP MENU (Paragraph C.2.3.2), after making any changes to the crontab file (supplementary user backups will not appear on this screen). Re-enter settings to the automatic backup schedule, if required.

Periodically check the contents of the generated files shown in Table C-5 to ensure that supplementary user backups are executing successfully. Results of the supplementary user backups are written to the file */tmp/bkup_user_perm_msg.txt* and mailed to the <*oradba*> Unix user account.

SECTION C.3 — BACKUP LIBRARY MANAGEMENT

C.3.1 OVERVIEW

Backup library management includes procedures for storage of backup tapes, tape marking and labeling conventions, and maintenance of a backup binder. These procedures are provided to assist with development of a comprehensive site backup and recovery plan, not to discourage site creativity. All procedures should be adapted to meet the unique requirements of individual sites. Careful planning and consistent management of backups can pay great dividends during database recovery.

C.3.2 TAPE ORGANIZATION

A well-organized tape storage library reduces the potential for extensive media loss or human error. During 24 hour operations, the possibility of human-error increases; standardized organization provides continuity and consistency even with multiple shifts.

The tape library should be situated in a well-lighted area. It should be maintained in a controlled environment and removed from risk of damage by liquids or accidental impacts.

As a minimum, the tape storage area should consist of three separate and independent compartments for full backups, cumulative backups and redo log backups. Physical distance between compartments is recommended to reduce the risk of error. Each compartment should be clearly marked with the type of backup.

Within each storage compartment, tapes should be stored with the oldest to the left and the most recent to the right. The full and cumulative backup compartments should be further subdivided and labeled for backup cycle week and day of the week (and AM/PM, if necessary). The number of weeks corresponds with the tape use cycle; this cycle should correspond with the automatic backup schedule. For example, full backups may be taken once a week, with tapes written over every two weeks. Such a cycle would require storage compartments for two weeks, with each day of the week clearly marked. It is advisable to use a storage door, or other device, to discourage access to tapes not in the current backup cycle week.

The redo log storage compartment should be subdivided into sections of 10, marked with the redo log tape labels. The number of sections varies from site to site, based on the number of redo log tapes required.

For each type of backup, a moveable marker may be used to mark the most recent backup.

Initially, the tape storage area should be labeled with temporary markings. It is very difficult to predict space requirements until site patterns are determined. Space must be allocated for heavier use during crisis operations, and database growth over time. Redo logs backups will experience the most growth during times of crisis, and may require additional tapes. It is advisable to periodically review and adjust tape storage areas to accommodate changes in site requirements.

C.3.3 LABELING CONVENTIONS

Clearly marked tape labels greatly reduce the potential for human error. Tapes should be labeled as soon as they are removed from their protective wrapping. An unlabeled tape should never be introduced into the system.

Tape labels should be marked both on the box and the tape, with a color corresponding to the type of backup. The tape label is a permanent label representing the type of backup and the day it was taken. Tapes should be labeled as designated in Table C-6.

Table C-6: Tape Labeling Conventions.

Backup Type	Color	Prefix Label	Suffix Label
REDO LOG BACKUP	Red	**REDO**	**- (*1-999*)**
FULL BACKUP	Black	**FULL*Tape_sequence***	***- Weekday AM/PM* WK*Week_cycle***
CUMULATIVE BACKUP	Green	**CUM*Tape_sequence***	**- *Weekday AM/PM* WK*Week_cycle***

The *tape_sequence* number is “1" for the first tape of the full or cumulative backup operation. If more than one tape is required, use “2", and so on. The *week_day* is the standard abbreviation for the day of the week (i.e., MON, TUE, WED). *AM/PM* is used at sites that require both an AM and PM full backup, and/or an AM and PM cumulative backup. The *week_cycle* is the number assigned to each week in the backup cycle (i.e., 1, 2). This number is the relative number representing each week of tape usage, not the calendar week.

The PREFIX LABEL is used by backup and restore programs to identify the contents of tapes in the drive. This PREFIX LABEL, stored as a label file at the beginning of each tape, helps prevent accidental overwriting of different backup types. It is also used in backup and restore programs to identify the correct sequence of tapes for multi-tape full and cumulative backups.

The SUFFIX LABEL is used as a tape library management tool. To allow site flexibility when tailoring backup cycles, it is not validated by backup and restore programs. Proper tape organization (see Paragraph C.3.2, Tape Organization) and backup binder maintenance (see Paragraph C.3.4, Backup Binder) contribute to recovery assurance in this area.

Each tape should be marked with the date of its entry into the system. This will assist in identification of problems arising from introduction of a new tape, and in replacement of worn-out media. In addition, sites with multiple client/servers should mark each tape with an identification of the server that the tape pertains to.

As an example, site X uses the standard backup schedule, with a tape use cycle of two weeks. Full backups are taken once a week on Monday, and Cumulative backups are taken on Tuesday, Wednesday, Thursday and Friday. Site X uses a 10-tape cycle for redo log backups. A full backup requiring two tapes is taken during the **first** week of the backup cycle. The backup tapes used for this operation are labeled with black ink: **FULL1 - MON WK1** and **FULL2 - MON WK1**. A Cumulative backup requiring one tape is taken on Wednesday of the **second** week of the backup cycle. The backup tape used for this operation is labeled with green ink: **CUM1 - WED WK2**. Redo log backups are taken whenever necessary. Redo log tapes are used consecutively as each tape fills. The first redo log tape in the cycle is marked with red ink: **REDO - 1**. The last redo log tape in the cycle is marked with red ink: **REDO - 10**.

Also see Paragraph C.2.9, Structural Change Backup.

C.3.4 BACKUP BINDER

The backup binder provides reference information about each backup to all administrators at a site. It is the definitive source for questions regarding any backup performed at the site within the tape use cycle. During recovery, the backup binder provides a quick and accurate way to determine the tape on which recovery data resides.

The backup binder should be a large loose-leaf style notebook which allows easy insertion and removal of hardcopy printouts in each section. As a minimum, the backup binder should have four sections: BACKUP CALENDAR, FULL BACKUP, CUMULATIVE BACKUP and REDO LOG BACKUP. Backup sections should be clearly labeled with the same color used for the backup tape (see Paragraph C.3.3, Labeling Conventions).

The BACKUP CALENDAR should cover a period of at least one tape use cycle. All planned backups should be scheduled, with day and time of backup. After each backup is successfully completed, it should be marked off the calendar. An effective technique is to annotate the tape label(s) of each successful backup beside the calendar date.

The FULL BACKUP section should contain a backup summary report for each full backup. The summary report contains the type and date of the backup, and the completion status. If the backup completed successfully, it contains the names of all tablespaces and files. The FULL BACKUP section should be maintained for the rolling automatic backup cycle (i.e., until the full backup tapes are overwritten). The backup summary report is a printout of the exit_status file sent to the printer after the backup is complete. You should annotate the tape label(s) on which the backup resides at the bottom right corner of the report. Each backup should be certified with your initials. Finally, you should discard the summary reports from any tapes that have been overwritten in the backup process.

The CUMULATIVE BACKUP section should contain similar information as the FULL BACKUP section. All procedures are similar to those for the full backup.

The REDO LOG BACKUP section should contain backup summary reports for each redo log backup tape. REDO LOG BACKUP summary reports are produced for each backup session on the tape. Each report lists the logs written to a redo log tape during that backup session. The header of each page provides the first log sequence number on tape; **"Current starting sequence on tape is *XXX*"**. Each time you switch to the next redo log tape, you should write the tape label beside the first report header for that tape.

You will use this report entry to find required redo log sequence numbers during recovery. Each backup should be certified with your initials. Finally, you should discard the summary reports from any redo log tapes that have been overwritten in the backup process.

During recovery, you will be prompted to apply a redo log number. Your entries in the REDO LOG BACKUP section of the Backup Binder will allow you to find the redo logs you need.

C.3.5 PREPARING TAPES

Tapes inserted into the tape drive should either be new tapes, or those used previously for the same type of backup (either full, cumulative, or redo log). For example, when inserting the tape for FULL_1 with a previous tape used for the full backup, that used tape must also be a FULL_1. The same is true for the cumulative and redo log backups. For redo log backup tapes, an old redo log backup tape may only be overwritten if the previous tape experienced a tar input/output error, or reached the end of media.

It is possible to manually view the header information for files written to tape by the backup and recovery software. To do this, you will need to use a blocking factor of 112, and the no-rewind tape option. First, rewind the tape using <*mt rewind*>. Next, view the first file header on tape with a command such as <*tar -tvbf 112 /dev/rmt/0hn*> (to avoid inadvertently overwriting database files and corrupting the database, change only the device setting in the previous command (i.e., /dev/rmt/0hn)). You may view the second and subsequent file headers on tape by repeating the same command again.

If a used tape is inserted that was not previously written for backup and recovery, the program will try to identify it and most likely reject it, especially if <*tar*> was used to store files on that tape. Use the <*mt erase*> (or <*mt -f /dev/rmt/0hn erase*>) command to erase these tapes in order to reuse them. (If time is critical, it is possible to erase only the first part of the tape. Use <*mt erase*>, allow the tape to run for five minutes, then kill the process or eject the tape. Rewind the tape completely using <*mt rewind*> before inserting it in the drive.)

SECTION C.4 — DATA RESTORATION PROCEDURES

C.4.1 EXAMPLE RECOVERY

The GCCS backup and recovery software can be used to recover a complete database using local backup tapes. The following example recovery has been provided to help you better understand the backup and recovery restore procedures. The example recovery should not be used as a substitute for the instructions provided in Paragraphs C.4.2 through C.4.6, and Section C.5 of this document.

The following conventions apply to the example recovery:

- *lower case italic* represents all commands/text that must be entered by the user.
- *<braketed italic>* represents commands entered by the user on the command-line.
- The symbol *<ENTER>* represents pressing the Enter or Return key at system prompts.
- **“bold within quotations”** represents messages and prompts generated by the system.
- The system messages and user entries that are shown as they will appear on the computer screen, are enclosed in double-lined borders.
- Text outside of the double-lined borders is included to guide you through the example recovery.

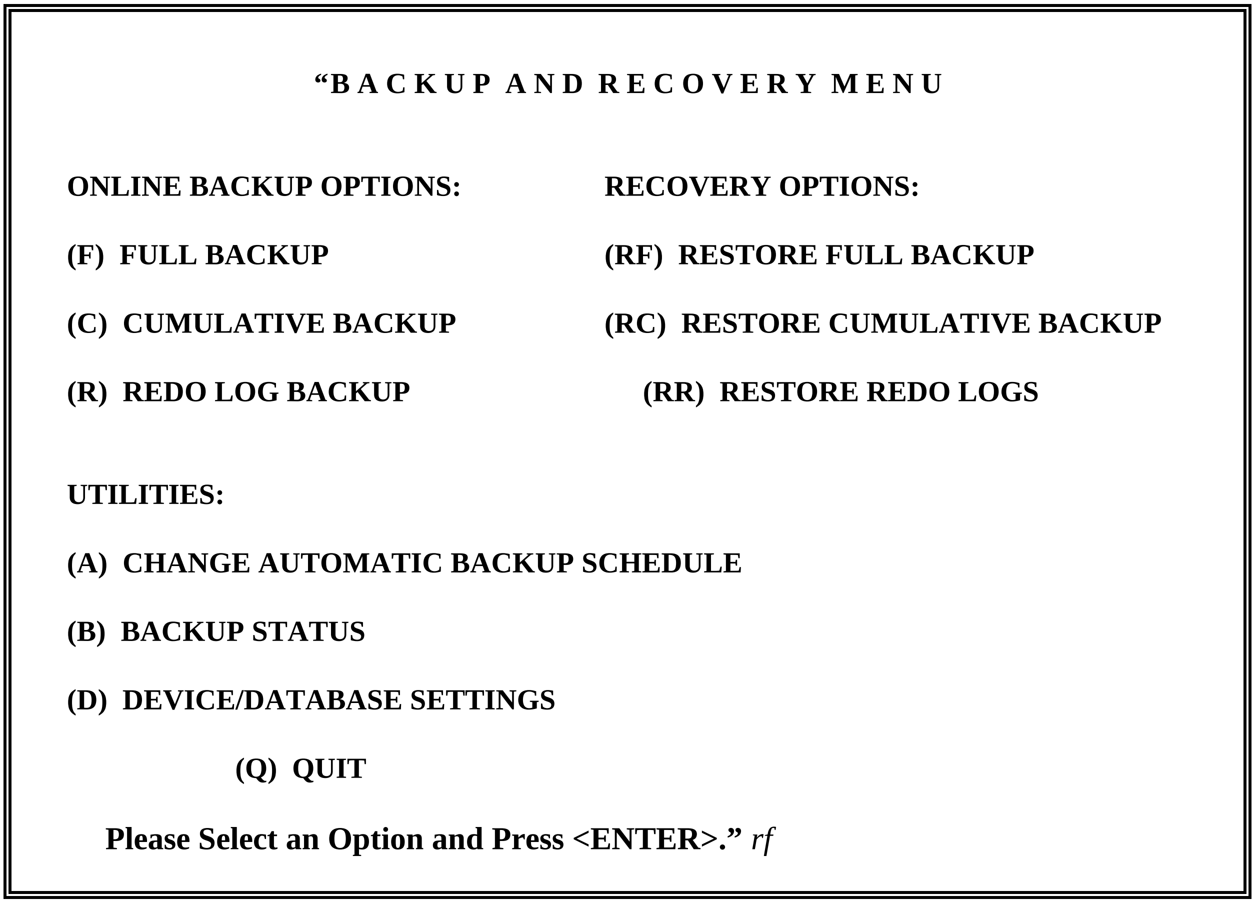
The example database is much smaller than the GCCS database. It consists of five tablespaces and eight data files. The ORACLE System Identification (SID) for the example database is GCCS1.

Log onto the database server as oradba. To begin database recovery, enter the *br_main* command at the Unix prompt (Figure C-9).

```
“$” <br_main>
```

Figure C-9: Starting the GCCS Backup and Recovery Program.

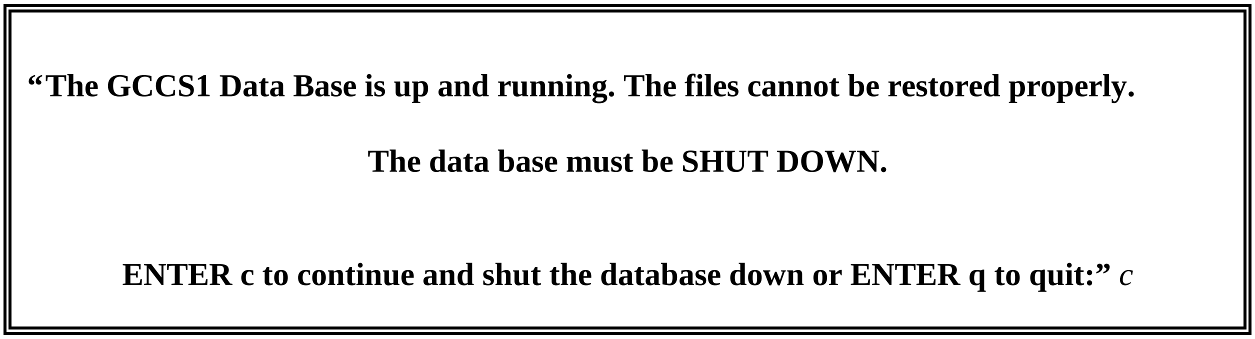
The <*br_main*> command pulls up the backup and recovery main menu (Figure C-10). Enter *rf* to start restoring from the full backup.



```
“BACKUP AND RECOVERY MENU

ONLINE BACKUP OPTIONS:           RECOVERY OPTIONS:

(F)  FULL BACKUP                 (RF)  RESTORE FULL BACKUP

(C)  CUMULATIVE BACKUP           (RC)  RESTORE CUMULATIVE BACKUP

(R)  REDO LOG BACKUP               (RR)  RESTORE REDO LOGS


UTILITIES:

(A)  CHANGE AUTOMATIC BACKUP SCHEDULE

(B)  BACKUP STATUS

(D)  DEVICE/DATABASE SETTINGS

          (Q)  QUIT

  Please Select an Option and Press <ENTER>.” rf
```

Figure C-10: RESTORE FULL BACKUP Selected from the Main Menu.

The full restore cannot take place while the database is online. If the database is online the message below is displayed (Figure C-11). Enter *c* to shut down the database. Figure C-12 displays a successful shutdown message. The message scrolls by quickly so you may not see it. Scroll backwards, if you would like to check the message.



```
“The GCCS1 Data Base is up and running. The files cannot be restored properly.

                    The data base must be SHUT DOWN.


     ENTER c to continue and shut the database down or ENTER q to quit:” c
```

Figure C-11: Shutting Down the Example Database in Preparation for the Full Restore.

```
"Mon 19:19> The GCCS1 Database is in the process of shutting down.



SQL*DBA: Release 7.1.3.0.0 - Production on Mon May 27 19:19:44 1996

Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.

Oracle7 Server Release 7.1.3.0.0 - Production Release
with the distributed option
PL/SQL Release 2.1.3.0.0 - Production

SQLDBA> Connected.
SQLDBA> Database closed.
Database dismounted.
ORACLE instance shut down.
SQLDBA> SQL*DBA complete."
```

Figure C-12: System Messages Indicating a Successful Database Shutdown.

If the database had been offline when you selected the RESTORE FULL BACKUP option from the main menu, the message below would have been the first message displayed (Figure C-13). Refer to the backup binder (Paragraph C.3.4, Backup Binder) to obtain the name of the most recent full backup tape labeled FULL_1. Place the tape into the proper tape drive, and enter *c* to continue the full restore.

```
        "You selected the Full Restore option.

        Insert the most recent tape labeled FULL_1


             Enter c to CONTINUE or q to QUIT:" c
```

Figure C-13: Continuing Full Restore Process After Mounting Correct Full Backup Tape.

The backup and recovery software identifies the tape, and begins restoring the database files to the disk (Figure C-14).

```
“Mon 19:19 > Searching for the FULL_1 header on tape.

    Mon 19:21 > Found the FULL_1 header on tape.

Mon 19:21> Restoring database file from tape.”
```

Figure C-14: System Messages Indicating a Successful Tape Identification.

As the files are copied from the tape to the disk, the progress of the program is displayed on the screen (Figures C-15, C-16, and C-17).

```
                “----------------------------------------------------------------------
                FILES CURRENTLY BEING RESTORED FROM TAPE
                ----------------------------------------------------------------------

x /tmp/APPL, 35 bytes, 1 tape blocks
x /h/SMDB/data/sm1/appl1.dbf, 1050624 bytes, 2052 tape blocks

Mon 19:21> Restoring database file from tape.”
```

Figure C-15: System Messages Showing Files Being Restored From Full Backup Tape Part I.

```
               “-----------------------------------------------------------------------
                FILES CURRENTLY BEING RESTORED FROM TAPE
                ----------------------------------------------------------------------

x /tmp/APPL, 35 bytes, 1 tape blocks
x /h/SMDB/data/sm1/appl1.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/DATA, 70 bytes, 1 tape blocks
x /h/SMDB/data/sm1/data1.dbf, 1050624 bytes, 2052 tape blocks
x /h/SMDB/data/sm1/data2.dbf, 1050624 bytes, 2052 tape blocks

Mon 19:21> Restoring database file from tape.”
```

Figure C-16: System Messages Showing Files Being Restored From Full Backup Tape Part II.

```
          “---------------------------------------------------------------------------
           FILES CURRENTLY BEING RESTORED FROM TAPE
           ---------------------------------------------------------------------------

x /tmp/APPL, 35 bytes, 1 tape blocks
x /h/SMDB/data/sm1/appl1.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/DATA, 70 bytes, 1 tape blocks
x /h/SMDB/data/sm1/data1.dbf, 1050624 bytes, 2052 tape blocks
x /h/SMDB/data/sm1/data2.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/RBSEGS, 37 bytes, 1 tape blocks
x /h/SMDB/data/sm1/rbsegs1.dbf, 1050624 bytes, 2052 tape blocks

Mon 19:21> Restoring database file from tape.”
```

Figure C-17: System Messages Showing Files Being Restored From Full Backup Tape Part III.

Once all of the data files from the full backup are restored to the disk, the tape is rewound and ejected (Figure C-18).

```
“Mon 19:22> Rewinding the FULL_1 backup tape.”
```

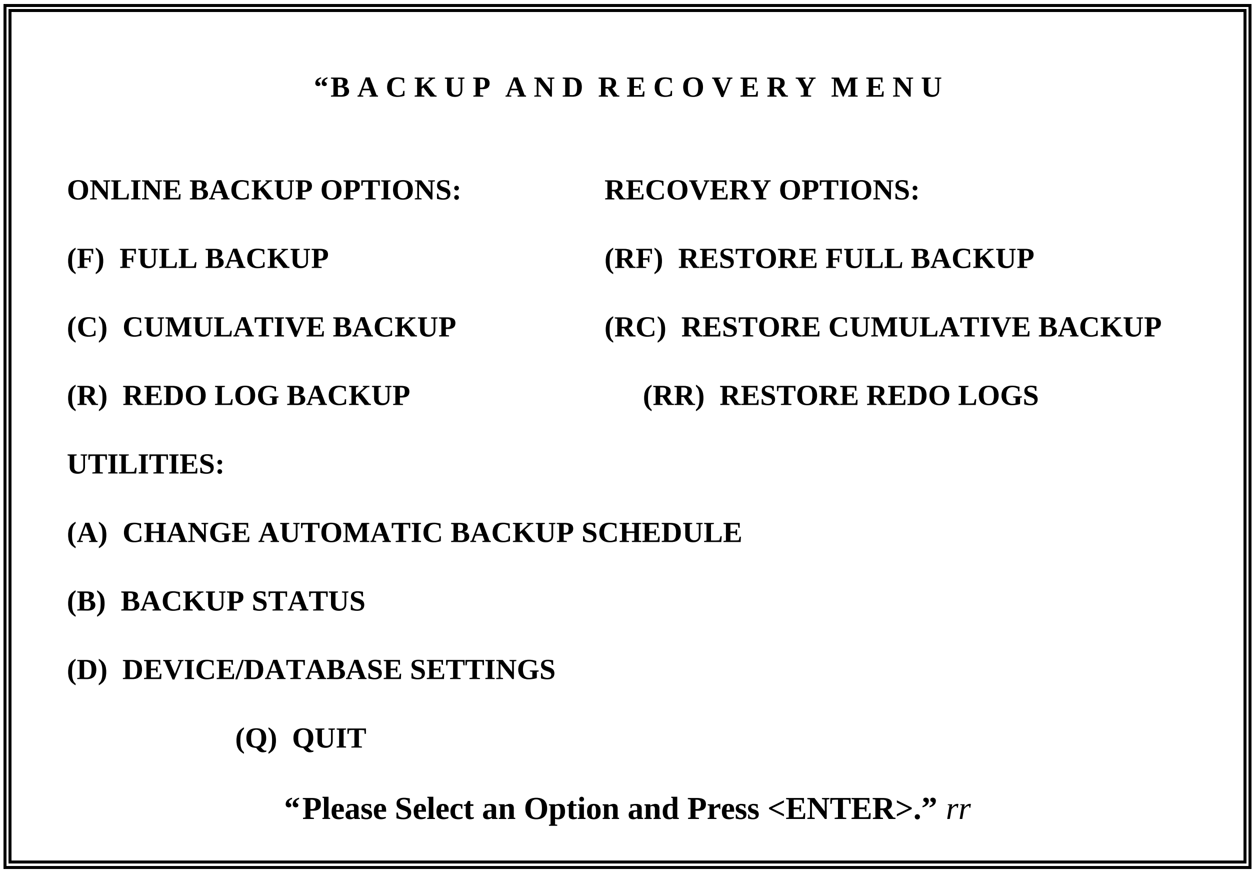
Figure C-18: Full Backup Tape Message.

After the tape is ejected, the exit status is printed and displayed (Figure C-19). “SPARC10" is the name of the printer used in this example (Paragraph C.2.2, Device/Database Settings). The exit status information is stored in the **/h/COTS/RDBMS/RECOVERY/restore/full/exit_status** file. To acknowledge the message and return to the main menu, press the return key.

```
“request id is SPARC10-19237 (1 file(s))

“***************************************************************************

     Mon May 27 19:22:35 EDT 1996

     The Full Restore program was SUCCESSFUL!

****************************************************************************

       The following files where restored from tape:
       x /tmp/APPL, 35 bytes, 1 tape blocks
x /h/SMDB/data/sm1/appl1.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/DATA, 70 bytes, 1 tape blocks
x /h/SMDB/data/sm1/data1.dbf, 1050624 bytes, 2052 tape blocks
x /h/SMDB/data/sm1/data2.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/RBSEGS, 37 bytes, 1 tape blocks
x /h/SMDB/data/sm1/rbsegs1.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/SYSTEM, 108 bytes, 1 tape blocks
x /h/SMDB/data/sm1/system1.dbf, 2099200 bytes, 4100 tape blocks
x /h/SMDB/data/sm1/system2.dbf, 1050624 bytes, 2052 tape blocks
x /h/SMDB/data/sm1/system3.dbf, 1050624 bytes, 2052 tape blocks
x /tmp/TEMP, 35 bytes, 1 tape blocks
x /h/SMDB/data/sm1/temp1.dbf, 1050624 bytes, 2052 tape blocks
x control.bak, 51712 bytes, 101 tape blocks
x initGCCS1.ora, 3705 bytes, 8 tape blocks
x full.dmp, 19456 bytes, 38 tape blocks

          Press <ENTER> to acknowledge this message:” <ENTER>
```

Figure C-19: Example Full Restore Exit Status Message.

Enter *rr* to begin the redo log restore process (Figure C-20).



```
                    "BACKUP AND RECOVERY MENU

ONLINE BACKUP OPTIONS:            RECOVERY OPTIONS:

(F)  FULL BACKUP                  (RF)  RESTORE FULL BACKUP

(C)  CUMULATIVE BACKUP            (RC)  RESTORE CUMULATIVE BACKUP

(R)  REDO LOG BACKUP                (RR)  RESTORE REDO LOGS

UTILITIES:

(A)  CHANGE AUTOMATIC BACKUP SCHEDULE

(B)  BACKUP STATUS

(D)  DEVICE/DATABASE SETTINGS

             (Q)  QUIT

          "Please Select an Option and Press <ENTER>." rr
```

Figure C-20: RESTORE REDO LOGS Selected from the Main Menu.

At this point the program tells you what commands you should enter when you are presented with the SQLDBA prompt (Figure C-21). After you enter the commands the system may return the message **"Statement processed"**. This would mean that all of the online and archived redo log files needed to recover the database were available on disk in the archive directory **/oracle/smback/arch**.

For the purposes of this example the required archived redo log files will not be available on disk. Enter the commands *CONNECT INTERNAL*, *STARTUP MOUNT*, and *ALTER DATABASE RECOVER AUTOMATIC*. When ORACLE tries to recover the database it will fail, and print a series of error messages. Read the ORACLE errors and make a note of the log sequence number required by ORACLE. In this case it is 727.

```
“Mon 19:26 > To begin the Redo log restoration process enter the following

        CONNECT INTERNAL <ENTER>
        STARTUP MOUNT <ENTER>
        ALTER DATABASE RECOVER AUTOMATIC;  <ENTER>

    at the SQLDBA> prompt.

SQL*DBA: Release 7.1.3.0.0 - Production on Mon May 27 19:26:23 1996

Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.

Oracle7 Server Release 7.1.3.0.0 - Production Release
With the distributed option
PL/SQL Release 2.1.3.0.0 - Production

SQLDBA>” CONNECT INTERNAL
“Connected”.
“SQLDBA>” STARTUP MOUNT
“ORACLE instance started.
Database mounted”.
“SQLDBA>” ALTER DATABASE RECOVER AUTOMATIC;
“ORA-00279: Change 25418 generated at 05/27/96 18:51:10 needed for thread 1
ORA-00289: Suggestion : /oracle/smback/arch/GCCS1_727.log
ORA-00280: Change 25418 for thread 1 is in sequence #727
ORA-00278: Logfile '/oracle/smback/arch/GCCS1_727.log' no longer needed for this
recovery
ORA-00308: cannot open archived log '/oracle/smback/arch/GCCS1_727.log’
ORA-07360: sfifi: stat error, unable to obtain information about file.
SVR4 Error: 2: No such file or directory”
```

Figure C-21: System Messages Indicating Correct User Input and Redo Log File Needed for Recovery.

Enter the *exit* command to leave the SQLDBA environment (Figure C-22).

```
“SQLDBA>” exit
“SQL*DBA complete.”
```

Figure C-22: Leaving The SQLDBA Environment.

Once you leave the SQLDBA environment control returns to the backup and recovery software. The system requests the log number necessary to complete the recovery process. Enter *727* to continue (Figure C-23).

```
“Enter the REDO LOG SEQUENCE number given by Oracle or q to QUIT:” 727
```

Figure C-23: Providing the Program with the Required Redo Log File Sequence Number.

Refer to the backup binder to obtain the label of the tape containing the redo log file GCCS1_727.log. Use of the backup binder to identify the required tape could save hours of tape search time. Place the tape in the proper tape drive, and enter *c* to continue (Figure C-24).

```
“The Restore Redo Log process needs log sequence numbers 727 to 741 from tape.

Please make sure the correct Redo Log backup tape is in place.

Enter c to CONTINUE or q to QUIT:” c
```

Figure C-24: Continuing Redo Log Restore Process After Mounting Correct Redo Log Backup.

The backup and recovery software identifies the tape (Figure C-25). Then it begins restoring redo log files to the archive directory beginning with GCCS1_727.log and ending with the most recently backup up redo log file. This could require the use of more than one redo log backup tape. However, in this example the starting log and the ending log are both on the same tape (Figure C-26).

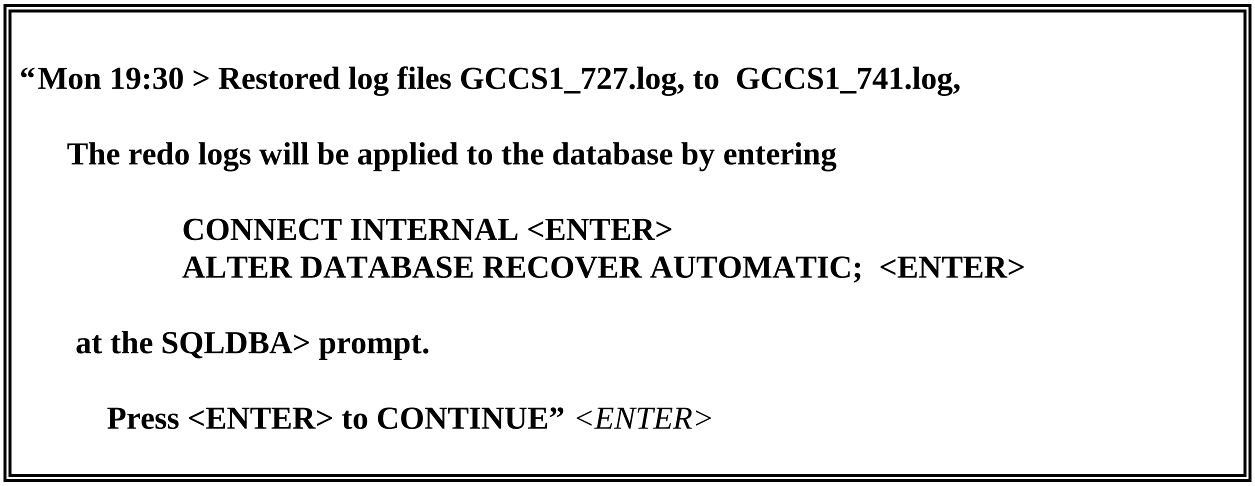
```
“Mon 19:28 > Searching for the REDO_LOG header on tape.

Mon 19:29 > Found the REDO_LOG header on tape.

Mon 19:29 > Searching for the log sequence 727 on tape.

Mon 19:30 > Found the log sequence 727 on tape.
        Repositioning tape to begining of file”
```

Figure C-25: System Messages Indicating a Successful Tape Identification.

```
“Mon 19:30 > Restored log file: x GCCS1_727.log, 26624 bytes, 52 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_728.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_729.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_730.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_731.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_732.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_733.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_734.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_735.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_736.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_737.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_738.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_739.log, 51712 bytes, 101 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_740.log, 23040 bytes, 45 tape blocks.

Mon 19:30 > Restored log file: x GCCS1_741.log, 51712 bytes, 101 tape blocks.”
```

Figure C-26: System Messages Showing Redo Log Files Being Restored from Redo Log Backup Tape.

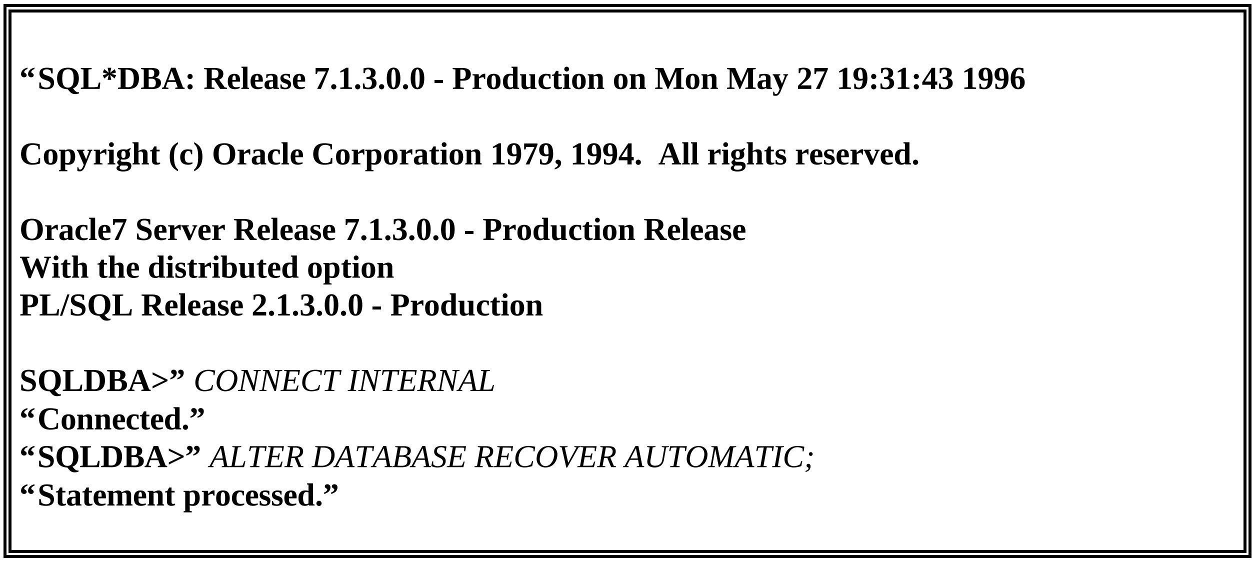
At this point the program tells you what commands you should enter when you are presented with the SQLDBA prompt (Figure C-27). The commands will remain on the screen so there is no need for you to memorize them or write them down. When you are ready to enter the commands, press the return key.



```
"Mon 19:30 > Restored log files GCCS1_727.log, to  GCCS1_741.log,

   The redo logs will be applied to the database by entering

            CONNECT INTERNAL <ENTER>
            ALTER DATABASE RECOVER AUTOMATIC;  <ENTER>

    at the SQLDBA> prompt.

      Press <ENTER> to CONTINUE" <ENTER>
```

Figure C-27: Continuing Recovery Process After Noting Instructional System Message.

Now that the logs have been copied from the tape to the archive directory, the ORACLE command *ALTER DATABASE RECOVER AUTOMATIC* will succeed. The message **"Statement processed"** indicates all required redo logs have been applied (Figure C-28).



```
"SQL*DBA: Release 7.1.3.0.0 - Production on Mon May 27 19:31:43 1996

Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.

Oracle7 Server Release 7.1.3.0.0 - Production Release
With the distributed option
PL/SQL Release 2.1.3.0.0 - Production

SQLDBA>" CONNECT INTERNAL
"Connected."
"SQLDBA>" ALTER DATABASE RECOVER AUTOMATIC;
"Statement processed."
```

Figure C-28: System Message Indicating Correct User Input and Successful Application of All Required Redo Logs.

Now the database is consistent. However, the database must be opened so that the general users can access the data. Open the database by entering the command *ALTER DATABASE OPEN* (Figure C-29).

```
SQLDBA>" ALTER DATABASE OPEN;
"Statement processed."
```

Figure C-29: Opening the Recovered Example Database.

At this point, the database has been recovered. Enter *exit* to leave SQLDBA (Figure C-30).

```
"SQLDBA>" exit
"SQL*DBA complete".
```

Figure C-30: Leaving the SQLDBA Environment.

After you leave SQLDBA control is returned to the backup and recovery software. Now the archive directory must be cleaned up to make room for new redo logs. Enter *c* to continue and the old logs will be removed automatically (Figure C-31).

```
  "The restored Redo Logs files GCCS1_727.log to GCCS1_741.log

            must removed from the hard drive.

        If you CONTINUE those files will be removed !!!!!

         Enter c to CONTINUE or q to QUIT:" c
```

Figure C-31: Cleaning Up the Archive Directory to Create Space for New Redo Logs.

Once the redo logs are removed the backup and recovery software may check to see if any archived redo logs were moved to other directories in an emergency situation. This function remains even though contingency redo log relocation is disabled in the existing software (Figure C-32).

```
GCCS1* :  No such file or directory

"Mon 19:32 > Searching for REDO LOGS in locations outside of

            the /oracle/smback/arch destination."
```

Figure C-32: Inconsequential Messages that May be Generated by the System.

After the backup and recovery software completes its cleanup activities, the redo log backup tape is rewound (Figure C-33).

```
“Mon 19:32> Rewinding the REDO LOG backup tape.”
```

Figure C-33: Redo Log Backup Tape Rewind.

Once the tape is rewound, the exit status of the redo log restore is displayed and printed. This information can be found in the **/h/COTS/RDBMS/RECOVERY/restore/redo_log/exit_status** file (Figure C-34).

```
“*****************************************************************************

    Mon May 27 19:32:56 EDT 1996

      The Redo Log Restore program was SUCCESSFUL!

******************************************************************************

   Archive log sequence numbers 727 to 741

   were restored to the archive directory.”
```

Figure C-34: Example Redo Log Restore Exit Status Message.

The message below, is a result of the print process (Figure C-35).

```
“request id is SPARC10-19239 (1 file(s))”
```

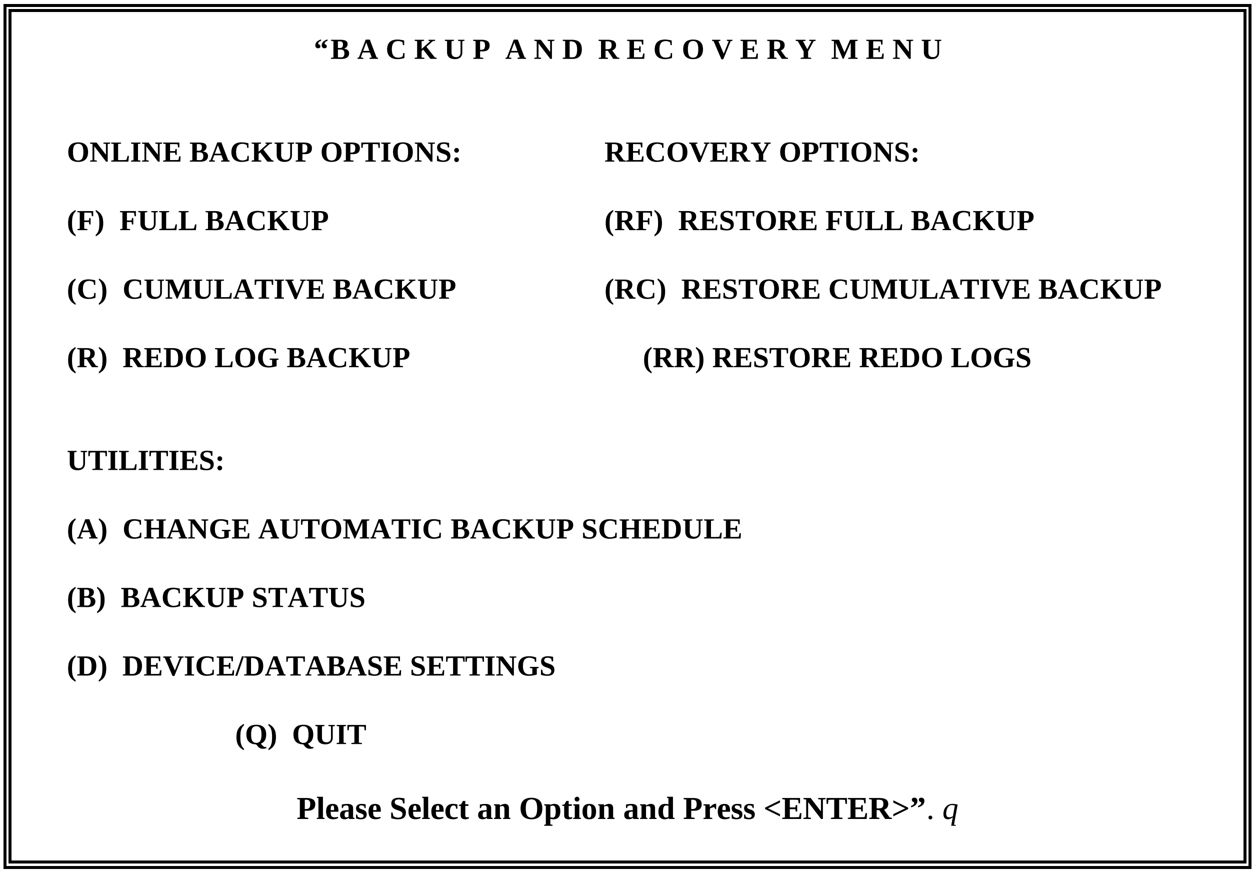
Figure C-35: System Message Indicating Successful Print.

Press the return key to acknowledge the exit status of the redo log restore program, and return to the backup and recovery main menu (Figure C-36).

```
“Press ENTER to acknowledge the message:” <ENTER>
```

Figure C-36: Continuing With the Post-Recovery Process.

Enter *q* to leave the backup and recovery main menu (Figure C-37).



```
                    "BACKUP AND RECOVERY MENU

ONLINE BACKUP OPTIONS:              RECOVERY OPTIONS:

(F)  FULL BACKUP                    (RF)  RESTORE FULL BACKUP

(C)  CUMULATIVE BACKUP              (RC)  RESTORE CUMULATIVE BACKUP

(R)  REDO LOG BACKUP                  (RR) RESTORE REDO LOGS


UTILITIES:

(A)  CHANGE AUTOMATIC BACKUP SCHEDULE

(B)  BACKUP STATUS

(D)  DEVICE/DATABASE SETTINGS

          (Q)  QUIT

             Please Select an Option and Press <ENTER>". q
```

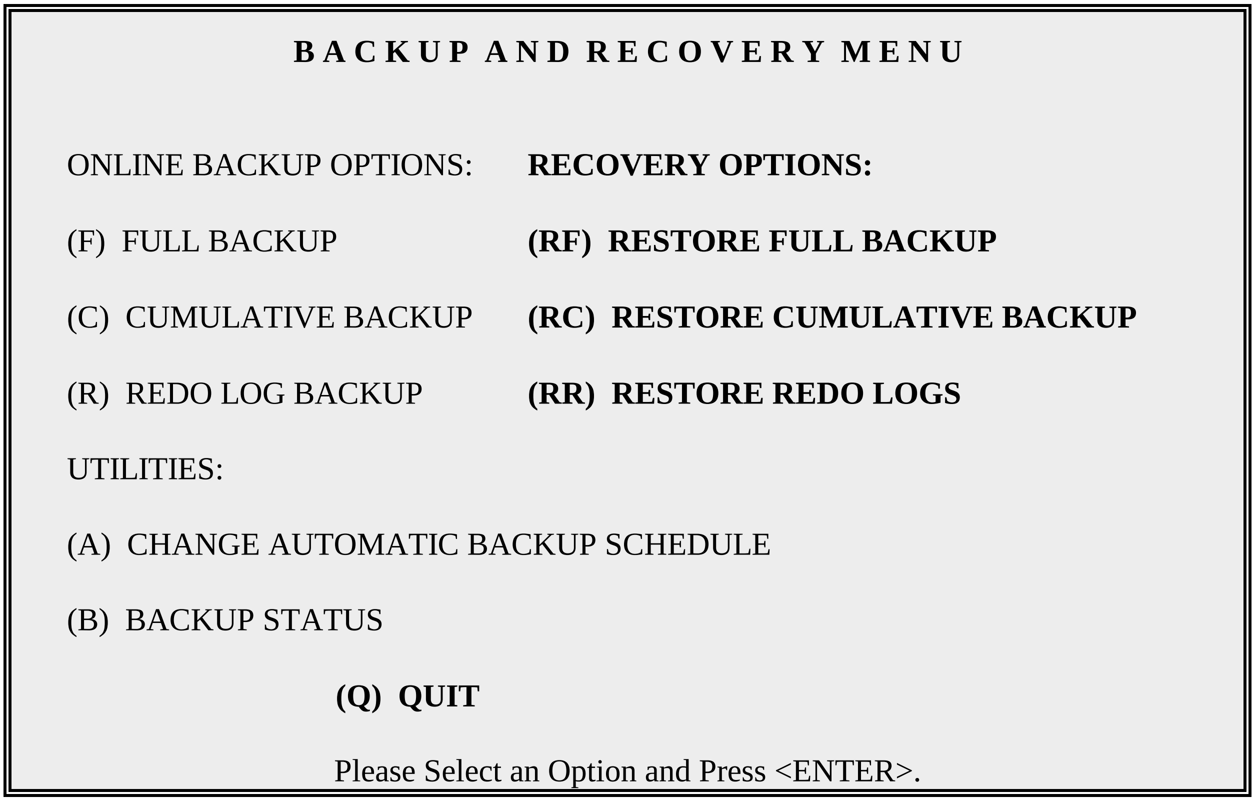
Figure C-37: Example Recovery Complete/Leaving the Main Menu.

C.4.2 BACKUP AND RECOVERY MENU: RESTORE OPTIONS

Restore procedures must be initiated from the BACKUP AND RECOVERY MENU.

Accessing backup and recovery software - To access backup and recovery software, log onto the database server directly from the GCCS “globe” as the Unix user <*oradba*> (do not *su* from another account). By default, you should be in the home directory of the <*oradba*> Unix account. Enter <*br_main*> to pull up the BACKUP AND RECOVERY MENU screen.

Figure C-38 shows the recovery options available from the BACKUP AND RECOVERY MENU.



```
                    BACKUP AND RECOVERY MENU

ONLINE BACKUP OPTIONS:     RECOVERY OPTIONS:

(F)  FULL BACKUP           (RF)  RESTORE FULL BACKUP

(C)  CUMULATIVE BACKUP     (RC)  RESTORE CUMULATIVE BACKUP

(R)  REDO LOG BACKUP       (RR)  RESTORE REDO LOGS

UTILITIES:

(A)  CHANGE AUTOMATIC BACKUP SCHEDULE

(B)  BACKUP STATUS

                (Q)  QUIT

                Please Select an Option and Press <ENTER>.
```

Figure C-38: Backup and Recovery Menu — Recovery Options.

To leave the BACKUP AND RECOVERY MENU, enter (Q) QUIT. You will return to the default oracle home directory.

All restore procedures are designed to complement ORACLE commands and procedures. Restore procedures are illustrated in Table C-7.

Table C-7: Restore Procedures.

PROCEDURE	BACKUP & RCVY MENU	AUTOMATIC BACKUP	PARAGRAPH
Restore full backup	YES	NO	C.4.2
Restore cumulative backup	YES	NO	C.4.3
Restore redo log backup	YES	NO	C.4.4
Recover control file	NO	NO	C.4.5

Although media recovery is performed on a case by case basis, the sequence in which files are restored from tape remains the same. A recovery incident may require one or more of the steps in the sequence described below. Advanced database administrators (knowledgeable in both unix tape commands and database procedures) will be able to save time during recovery by performing isolated tablespace recovery without recovering the entire database.

First, the most recent full backup is restored. The most recent cumulative backup is restored next, overwriting any files that have changed since the full backup. Each of these backup scripts writes data files from tape directly to the appropriate path on disk. (The ORACLE import, control file, and init.ora file are written to a staging area on disk, for use only if required. These files are written to the following directories during database restores: **$ORACLE_HOME/RECOVERY/restore/full** and **$ORACLE_HOME/RECOVERY/ restore/cum**).

The last step in the recovery process is application of redo logs to roll the database forward to the present point in time. ORACLE prompts for the first redo log required. The site administrator loads the correct redo log tape and uses the restore redo log program to move the logs from tape to the backup disk. ORACLE applies redo logs sequentially from the backup disk until the database has been rolled forward. The backup disk may fill before all necessary logs are moved from tape; the redo log process will allow you to alternate between applying logs to roll the database forward, and restoration of logs from tape to the backup disk.

Many possibilities exist for media recovery. In many cases, only the most recent archived redo logs on the backup disk and/or the online redo logs will be required. In other cases, files that require recovery may all be contained in the last cumulative backup (refer to the Backup Binder to make this determination). In any case, the administrator will need all archived redo logs to the point of the earliest restored full backup. Refer to Section C.5, Database Recovery, for more information.

C.4.3 RESTORE FULL BACKUP

The restore full backup procedure is initiated manually from the BACKUP AND RECOVERY MENU. The database should be offline while the restore procedure is executing. First, refer to the backup binder to ensure that the correct full backup tape is inserted in the tape drive. Next, select (RF) RESTORE FULL BACKUP from the BACKUP AND RECOVERY MENU.

The restore full backup procedure runs in the foreground and moves all tablespaces from the backup tape to their original directories in the ORACLE database. Data files on tape overwrite any existing data files. As each file is restored, it is displayed, by tablespace, on the console screen. The control file and init.ora file are restored to the **$ORACLE_HOME/RECOVERY/restore/full** directory, for use if required (i.e., the file is damaged, and mirrored or duplicate copies are unavailable). The full database export (no rows) is also restored to this directory, ready for an import command, if required.

C.4.3.1 Restore Full Backup Execution

Before the first tablespace is written to disk, the procedure reads the first label file on tape and confirms that the tape is the first tape of a full backup operation. Next, each tablespace (stored as a tarfile) is read sequentially into the database. After all tablespaces are copied, the control file is moved to the restore directory, stored as **$ORACLE_HOME/RECOVERY/ restore/full/control.bak**. Next, the init.ora file is moved to the restore directory, stored as **$ORACLE_HOME/RECOVERY/restore/full/init.ora**. As the last step of the restore, the export file (structures and system data) is moved to the restore directory, stored as **$ORACLE_HOME/RECOVERY/restore/full/full.dmp**. (The file **full.dmp** is in ORACLE export format, ready for the ORACLE import command).

If more than one tape is required for the full restore, insert the next tape in sequence when prompted. Press <*enter*> to continue the restore operation.

Only one backup or restore operation should ever run at one time. Before execution, the full restore checks if ORACLE is up, or if any other processes are running, and provides appropriate error messages. During execution, the full restore program checks for any other backup or restore processes currently running, and establishes a lock. This procedure prevents errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the full restore program. Warning errors generally occur after all tablespaces have been moved to disk and include errors encountered during control file or export restoration. Most errors are terminating errors and include write and permission errors, tar I/O errors, and locking errors (another backup or restore is currently running). All errors are written to two files: the **error.log** (accumulation of all error messages) and **exit_status** (restore summary log sent to screen and printer).

In the event of program termination you must correct the problem and manually initiate the full restore again from the BACKUP AND RECOVERY MENU. Files successfully written to disk before a terminating error will not be deleted; they will be overwritten when the program is re-executed. Warning errors are less critical errors that should be investigated, but may not require restarting the full restore manually (dependent on the extent of media damage and/or condition of cumulative restore files).

C.4.3.2 Restore Full Backup Completion

After execution of the full restore program, a restore summary log is sent to the screen and printer. This log contains the type of restore, the date and time, and the final status of the program (SUCCESSFUL or TERMINATED). The report will also list the files copied to disk, grouped by tablespace name. All files successfully copied to disk will be included, regardless of the final status of the restore operation. You should keep this report until recovery is complete.

C.4.4 RESTORE CUMULATIVE BACKUP

The restore cumulative backup procedure is initiated manually from the Backup and Recovery menu. The database should be offline while the restore procedure is executing. First, refer to the backup binder to ensure that the correct cumulative backup tape is inserted in the tape drive. Next, select (RC) RESTORE CUMULATIVE BACKUP from the BACKUP AND RECOVERY MENU.

The restore cumulative backup procedure runs in the foreground, and moves all tablespaces from the backup tape to their original directories in the ORACLE database. Data files on tape overwrite any existing files in cumulative backup tablespaces. As each file is restored, it is displayed, by tablespace, on the console screen. The control file is restored to the **$ORACLE_HOME/RECOVERY/restore/cum** directory, for use if required (i.e., the file is damaged, and mirrored or duplicate copies are unavailable).

C.4.4.1 Restore Cumulative Backup Execution

Before the first tablespace is written to disk, the procedure reads the first label file on tape and confirms that the tape is the first tape of a cumulative backup operation. Next, each tablespace (stored as a tarfile) is read sequentially into the database. After all tablespaces are copied, the control file is moved to the backup disk, stored as **$ORACLE_HOME/RECOVERY/restore/cum/control.bak**.

If more than one tape is required for the cumulative restore, insert the next tape in sequence when prompted. Press <*enter*> to continue the restore operation.

Only one backup or restore operation should ever run at one time. Before execution, the cumulative backup checks if ORACLE is up, or if any other processes are running, and provides appropriate error messages. During execution, the cumulative restore program checks for any other backup or restore processes currently running, and establishes a lock. This procedure prevents errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the cumulative restore program. All errors are terminating errors and include write and permission errors, tar I/O errors, and locking errors (another backup or restore is currently running). All errors are written to two files: the **error.log** (accumulation of all error messages) and **exit_status** (restore summary log sent to screen and printer).

In the event of program termination you must correct the problem and manually initiate the cumulative restore again from the Backup and Recovery menu. Files successfully written to disk before a terminating error will not be deleted; they will be overwritten when the program is re-executed.

C.4.4.2 Restore Cumulative Backup Completion

After execution of the cumulative restore program, a restore summary log is sent to the screen and printer. This log contains the type of restore, the date and time, and the final status of the program (SUCCESSFUL or TERMINATED). The report will also list the files copied to disk, grouped by tablespace name. All files successfully copied to disk will be included, regardless of the final status of the restore operation. You should keep this report until recovery is complete.

C.4.5 RESTORE REDO LOG BACKUP

The restore redo log backup procedure is initiated manually from the BACKUP AND RECOVERY MENU. It is used to restore archived redo logs and interactively apply them to roll the database forward. The restore redo log program should be executed after the most recent full and cumulative backups have been restored and you are ready to begin media recovery.

C.4.5.1 Restore Redo Log Backup Execution

First, shut down the ORACLE database. Select (RR) RESTORE REDO LOG BACKUP from the BACKUP AND RECOVERY MENU. (If the database is open, a message will be displayed. You may enter (C) CONTINUE, automatically shutting the database down, or (Q) QUIT to exit and return to the BACKUP AND RECOVERY MENU.)

Next, the program will provide a SQLDBA> prompt. You should enter *CONNECT INTERNAL*, then start up the database in recovery mode by using the command *STARTUP MOUNT*. (If you receive an error that the control file is not found, refer to Paragraph C.4.5, Recover Control File). Next, you should attempt to roll the database forward using the command *ALTER DATABASE RECOVER AUTOMATIC;*. If ORACLE begins to apply redo logs at this point, you do not need to restore logs from tape; skip to the **BACKUP DISK** section of Table C-8. If any other messages are returned, follow the steps in the next paragraph.

If ORACLE cannot find the log number it needs to begin recovery, it will return messages similar to ones in the example below:

“

```
ORA-00279: Change 11448 generated at 03/22/96 10:37:9 needed for thread 1
ORA-00289: Suggestion : /oracle/smback/arch/GCCS_1348.log
ORA-00280: Change 11448 for thread 1 is in sequence #1348
ORA-00278: Logfile ‘/oracle/smback/arch/GCCS_1348.log ‘ no longer needed for this recovery
ORA-00308: cannot open archived log ‘/oracle/smback/arch/GCCS_1348.log’
ORA-07360: sfifi: stat error, unable to obtain information about file.
SVR4 Error: 2: No such file or directory
```

“

In the example above ORACLE shows that log number 1348 is needed to start the recovery process. (The views **V$LOG_HISTORY** and **V$RECOVERY_LOG** (database mounted but not open) may also be used to determine the starting log requirement. After determining the required starting redo log number, enter <*exit*> to leave SQL*DBA and begin restoration of redo logs from tape.

After you have typed <*exit*> (see previous paragraph), the restore program will prompt for the redo log sequence number ORACLE requires to begin recovery. Enter the determined starting redo log number (e.g., '1348'). The program will attempt to find the location of all redo logs in sequence, from the starting log to the most recent. Skip to the **TAPE** section of Table C-8.

Note: The contingency redo log restore procedures described in the restore instructions are for general reference only. Under the current system configuration, the contingency redo log backup function will not execute (refer to Paragraph C.2.6, Perform Redo Log Backup).

If the starting log is located on a contingency drive (the backup disk was full, and the redo log backup program wrote redo logs to free space on other drives), the program will produce a report listing the location of redo logs on contingency drives; skip to the CONTINGENCY section of Table C-8.

If the starting log is located on tape, the program will wait for you to insert a redo log backup tape in the tape drive; skip to the TAPE section of Table C-8.

Table C-8 provides the recovery sequence for the redo log restore program. Follow the procedures section by section, starting with the section where the starting redo log resides.

Redo logs on tape overwrite any redo logs of the same name on disk.

Only one backup or restore operation should ever run at one time. Before execution, the restore redo log backup program checks for any other backup or restore processes currently running, and establishes a lock if none are found. This procedure prevents errors caused by simultaneous backup and/or recovery operations.

All routine errors are trapped by the archived redo log restore program. All archived redo log errors are terminating. Terminating errors include write and permission errors, tar I/O errors, and locking errors (another backup or restore is currently running). If some logs were successfully written from tape before the error occurred, you may start the next restore with the log number of the last partially written redo log. All errors are written to two files: the **error.log** (accumulation of all error messages) and **exit_status** (restore summary log sent to screen and printer).

In the event of program termination you must correct the problem and manually initiate the redo log restore again from the BACKUP AND RECOVERY MENU.

Table C-8: Restore Redo Log - Recovery Sequence.

RECOVERY SEQUENCE	PROCEDURE
TAPE	Refer to the backup binder (see Paragraph C.3.4, Backup Binder) to ensure that the redo log backup tape containing the **first log needed by ORACLE** is inserted in the tape drive. The restore program searches the tape to confirm that the starting log number entered is present on tape. Each redo log is copied sequentially into the default archive directory, starting with the entered log number and ending with the last log not already on the backup disk. While restoring redo log files from tape, all required logs may be copied to the backup disk, or the backup disk may run out of space. In either case, the following message appears: **" Restored log files x to y** **The redo logs will be applied to the database by entering** **CONNECT INTERNAL <ENTER>** **ALTER DATABASE RECOVER AUTOMATIC; <ENTER>** **at the SQLDBA> prompt"** Press ENTER to reach the SQLDBA prompt, and enter the commands shown above. Note that you must enter the commands shown above to apply the redo logs to the database; the restore program only copies the logs to the backup disk staging area. If all required logs have been applied, ORACLE will return WITHOUT an error that some redo logs could not be found. Skip to the BACKUP DISK section. If additional logs are required to complete recovery, ORACLE will detect a break in log sequence numbers and provide the message that a redo log was not found. Type <*exit*> to logout from SQL*DBA. The following message appears: **"The Restore Redo Log program still needs to restore log files starting with the file after redo_x.arc, up to the log sequence y."**

RECOVERY SEQUENCE	PROCEDURE
(TAPE)	You may select (Q) QUIT, or (C) CONTINUE the restoration process. If you select (C), the most recently restored redo logs will be removed from the backup drive. The following message appears: **"The restored Redo Log files x to y must be removed from the hard drive. If you no longer need those files, remove them now !!!"** **If you select (C) CONTINUE, the most recently applied logs will be deleted from the backup disk.** After this is complete, the program will write the next set of redo logs from tape to the backup disk. Repeat this process until all required logs have been restored to the backup disk and applied. If additional logs are located on a contingency drive (the backup disk was full, and the redo log backup program wrote redo logs to free space on other drives), ORACLE will return the message that a redo log was not found. Exit from SQL*DBA. The program will provide messages that it is searching for logs, then compiling the Redo Log Location Report. Next, it will produce a report listing the location of redo logs on contingency drives (stored as **$ORACLE_HOME/RECOVERY/restore/redo_log/log_location.rpt**) to the screen and printer. Skip to the CONTINGENCY section. If there are no logs on a contingency drive, the last successful *ALTER DATABASE RECOVER AUTOMATIC;* command will roll the database forward, applying all logs on the backup disk. Skip to the BACKUP DISK section.
CONTINGENCY	If logs required to roll the database forward are located on a contingency drive, refer to the "Redo Log Location Report". Determine the path of the redo log not found by ORACLE. The most recent logs will be located on the backup disk, with earlier logs already moved to contingency drives. The program will provide a SQLDBA> prompt. You should *CONNECT INTERNAL*, then issue the command listed on the "Redo Log Location Report" using the format *ALTER DATABASE RECOVER AUTOMATIC FROM <u>pathname</u>;*. If ORACLE applies a number of logs, then returns the message that a redo log was not found, repeat this process until all required redo logs have been located and applied. (Subsequent recovery commands within a SQL*DBA session will use the format *ALTER DATABASE RECOVER AUTOMATIC FROM <u>pathname</u> CONTINUE DEFAULT;*). After all redo logs have been applied from contingency drives, apply the most recent logs (all logs remaining on the backup disk). To apply the remaining logs, enter *ALTER DATABASE RECOVER AUTOMATIC CONTINUE DEFAULT;*. Skip to the BACKUP DISK section.

RECOVERY SEQUENCE	PROCEDURE
BACKUP DISK	If the *ALTER DATABASE RECOVER AUTOMATIC;* executed successfully WITHOUT returning the message that a redo log was not found, all redo logs on the backup disk were applied to roll the database forward. Skip to the RECOVERY COMPLETE section.
RECOVERY COMPLETE	After receiving the message **"STATEMENT PROCESSED"**, issue the command *ALTER DATABASE OPEN;*. (If you used a backup control file, you must open the database with the *RESETLOGS* option, and immediately perform an online full backup). When the command is complete, exit from SQL*DBA. The database has been rolled forward using all required archived and online redo logs. Exit from SQL*DBA. The following message appears: **"The restored Redo Logs files x to y must be removed from the hard drive. If you no longer need those files, remove them now !!!"** **If you select (C) CONTINUE, the most recently applied logs will be deleted from the backup disk.** The Backup and Recovery menu will appear. The ownership of the restored data (.dbf) files should be restored from owner <*oradba*> to owner <*oracle*>. To do this using the current software, follow the following procedure: From the <*oradba*> account, enter the command *sqlplus / @/h/COTS/RDBMS/scripts/chmod_dbf.sql* After completion of this script, prepare the generated file for execution by entering *chmod 777 /tmp/dbfs.csh* Next, login to the unix <*root*> account and enter the command */tmp/dbfs.csh* After completion of the script, delete the generated file by entering the command *rm /tmp/dbfs.csh* Logout of the unix <*root*> account.

C.4.5.2 Restore Redo Log Backup Completion

After execution of the redo log restore program, a restore summary log is sent to the screen and printer. This log contains the type of restore, the date and time, and the final status of the program (SUCCESSFUL, SUCCESSFUL with WARNINGS, or TERMINATED). The report will also list the range of redo log files copied to disk. All redo logs successfully copied to disk will be included, regardless of the final status of the restore operation. You should keep this report until recovery is complete.

Refer to Section C.5, Database Recovery, for more information on archived redo log recovery.

C.4.6 RECOVER CONTROL FILE

The full and cumulative restore procedures move a backup copy of the control file to the backup disk. The backup copies should only be used if the current control file and ALL mirrored copies are destroyed. Each mirrored control file in the database is an exact duplicate which may be copied to the mirrored location. Control files change as a result of creating or dropping a tablespace, adding, renaming (relocating) existing data files, or adding, renaming (relocating) online redo logs (Refer to Paragraph C.2.9, Structural Change Backup).

Wherever possible, directory paths provided in sample commands are accurate; if your site configuration has been changed, you may need to adjust one or more path specifications. In most cases, the variables *$ORACLE_SID* and *$ORACLE_HOME* may be entered as shown).

In most cases, you will discover that the current control file is missing or corrupted during the restore redo log backup procedure. After issuing the *STARTUP MOUNT* command at the first SQLDBA> prompt, you will receive an ORACLE error pertaining to the *ctrl1$ORACLE_SID.ctl* file. Table C-9 provides the recovery sequence to restore the control file.

Table C-9: Restore Control File - Recovery Sequence.

RECOVERY SEQUENCE	PROCEDURE
LOCATE MIRRORED CONTROL FILES	First, exit from SQL*DBA. Select (Q) QUIT to quit from the Redo Log Restore program. (**Note**: In the examples below, $ORACLE_SID is GCCS) At the system prompt, find the location of all control files using the Unix command: *<grep ctrl $ORACLE_HOME/dbs/config$ORACLE_SID.ora>* Skip to the RESTORE MIRRORED CONTROL FILE section.
RESTORE MIRRORED CONTROL FILE	Copy the first mirrored control file to the main control file location. For example: *<cp h/COTS/RDBMS/dbs/ctrl2$ORACLE_SID.ctl /h/COTS/RDBMS/dbs/ctrl1$ORACLE_SID.ctl >* Enter SQL*DBA by entering the Unix command: *<sqldba lmode=y>* You should receive the SQLDBA> prompt. Attempt to startup the database by entering the commands: *CONNECT INTERNAL* *STARTUP MOUNT* If the database instance starts, the control file has been successfully restored skip to the RESTORE BACKUP CONTROL FILE COMPLETE section. If you still receive an error that ORACLE cannot find **ctrl1*$ORACLE_SID*.ctl** or the control file is corrupted, you should try to restore the next mirrored control file. Skip to the LOCATE MIRRORED CONTROL FILES section to identify the next available control file. Repeat the steps in this section for the next available control file.

RECOVERY SEQUENCE	PROCEDURE
(RESTORE MIRRORED CONTROL FILE)	If you still receive an ORACLE error pertaining to the **ctrl1*$ORACLE_SID*.ctl** file and you have already attempted to restore all mirrored control files, you will need to restore the control file backup. Skip to the section RESTORE BACKUP CONTROL FILE.
RESTORE BACKUP CONTROL FILE	Exit from SQL*DBA. Next, copy the backup control file to the main control file location. Use the Unix command: *<cp $ORACLE_HOME/RECOVERY/restore/cum/control.bak h/COTS/RDBMS/dbs/ctrl1$ORACLE_SID.ctl>* Next, for consistency, copy the backup control file to all mirrored locations. Locate all mirrored control files using the Unix command: *<grep ctrl $ORACLE_HOME/dbs/config$ORACLE_SID.ora>* Copy the backup control file to each mirrored location. For example: *<cp h/COTS/RDBMS/dbs/ctrl1$ORACLE_SID.ctl h/COTS/RDBMS/dbs/ctrl2$ORACLE_SID.ctl>* Enter SQL*DBA by entering the Unix command: *<sqldba lmode=y>*
(RESTORE BACKUP CONTROL FILE)	You should receive the SQLDBA> prompt. Startup the database by entering the commands: *CONNECT INTERNAL* *STARTUP MOUNT* Next, get the current log sequence number using the command: *archive log list* Write the current log sequence number down; you will need it to perform incomplete media recovery of the database. Proceed to the section RESTORE BACKUP CONTROL FILE COMPLETE.

RECOVERY SEQUENCE	PROCEDURE
RESTORE BACKUP CONTROL FILE COMPLETE	If the database started up successfully, you are ready to restore redo logs and roll the database forward. Exit from SQL*DBA and run the RESTORE REDO LOGS option from the BACKUP AND RECOVERY MENU. The program will shut the database down as a precautionary measure. The Redo Log Restore program provides an SQLDBA> prompt. You should *CONNECT INTERNAL* and then start the database with the *STARTUP MOUNT* command. The database recovery process can now begin by using the command *RECOVER DATABASE USING BACKUP CONTROLFILE;*. If ORACLE returns a message indicating that recovery is not needed skip to the RECOVERY COMPLETE FULL section. If the database requires recovery the following example shows the results of the command. SQLDBA> *RECOVER DATABASE USING BACKUP CONTROLFILE;* **"ORA-00279: Change 19137 generated at 01/04/94 13:48:44 needed for thread 1** **ORA-00289: Suggestion : /usr2/oracle/logs/redo_1.arc** **ORA-00280: Change 19137 for thread 1 is in sequence #1** **Specify log: {<RET>=suggested | filename | AUTO | FROM logsource | CANCEL}** **Press <ENTER> to apply the suggested log file."** The following message will be displayed. **"Applying suggested logfile..."** The next suggested log file will be displayed. If the required log file is not found, type *cancel*, then *exit.* Enter the log number required, and restore all required logs. Apply the logs using the SQLDBA command provided above. CONTINUE APPLYING LOGS UNTIL THE SUGGESTED LOG SEQUENCE NUMBER EQUALS THE CURRENT LOG SEQUENCE NUMBER WHICH WAS OBTAINED FROM THE *ARCHIVE LOG LIST* command then exit out of the recovery process by entering *CANCEL*. Proceed to the RECOVERY COMPLETE CANCEL BASED section.

RECOVERY SEQUENCE	PROCEDURE
CONTINGENCY	Additional logs may be required to roll the database forward which are located on a contingency drive. Refer to the "Redo Log Location Report". To apply the contingency logs, determine the path of the redo log not found by ORACLE. The most recent logs will be located on the backup disk, with earlier logs already moved to contingency drives. The Redo Log Restore program provides a SQLDBA> prompt. You should *CONNECT INTERNAL* then the recovery process can be initiated by using the command *RECOVER DATABASE USING BACKUP CONTROLFILE;*. The recover command will indicate that it cannot find the redo log along with the following prompt on the last line. **"Specify log: {<RET>=suggested | filename | AUTO | FROM logsource | CANCEL}"** To apply the logs from the contingency drives enter *FROM* *pathname* using the pathname as given from the "Redo Log Location Report" on the line provided beneath the prompt then press <ENTER> for each of the suggested logs then continue to next contingency drive pathname. CONTINUE APPLYING LOGS UNTIL THE SUGGESTED LOG SEQUENCE NUMBER EQUALS THE CURRENT SEQUENCE NUMBER WHICH WAS OBTAINED FROM THE *ARCHIVE LOG LIST* command or all contingency logs are applied. If all the contingency logs are applied and the current log sequence number has not been reached then continue applying logs as shown in the RECOVER command example at the RESTORE BACKUP CONTROL FILE COMPLETE section. If the current log sequence has been applied then exit out of the recovery process by entering *CANCEL*. Proceed to the RECOVERY COMPLETE CANCEL BASED section.

RECOVERY SEQUENCE	**PROCEDURE**
TAPE	ORACLE will detect a break in log sequence numbers and provide the message that a redo log was not found. Exit from SQL*DBA. The Redo Log Restore program will prompt for the sequence number. Enter the number and press return. The redo logs will be extracted from tape. Upon completion of extracting the redo logs from tape the program provides a SQLDBA> prompt. The redo logs will be applied to the database by entering: *CONNECT INTERNAL* *RECOVER DATABASE USING BACKUP CONTROLFILE;* Follow the example of the *RECOVER* command as shown in the RESTORE CONTROL FILE COMPLETE section. While restoring redo log files from tape, the backup disk may run out of space. If this occurs, the following message appears: **"The Restore Redo Log program still needs to restore log files starting with the file after redo_x.arc, up to the log sequence y."** **You may select (Q) to quit, or (C) to continue the restoration process. Next, the following message appears:** **"The restored Redo Log files x to y must be removed from the hard drive. If you no longer need those files, remove them now !!!"** If you select (C) CONTINUE, the most recently applied logs will be deleted from the backup disk. After this is complete, the program will write the next set of redo logs from tape to the backup disk. Repeat this process until all required logs have been restored to the backup disk and applied.

RECOVERY SEQUENCE	PROCEDURE
(TAPE)	CONTINUE APPLYING LOGS UNTIL THE SUGGESTED LOG SEQUENCE NUMBER EQUALS THE CURRENT LOG SEQUENCE NUMBER WHICH WAS OBTAINED FROM THE ARCHIV *E LOG LIST* command. If additional logs are located on a contingency drive (the backup disk was full, and the redo log backup program wrote redo logs to free space on other drives), ORACLE will return the message that a redo log was not found. Exit from SQL*DBA. The program will provide messages that it is searching for logs, then compiling the Redo Log Location Report. Next, it will print a report listing the location of redo logs on contingency drives (stored as **$ORACLE_HOME/RECOVERY/restore/redo_log/log_location.rpt**) to the screen and printer. Skip to the CONTINGENCY section. If there are no logs on a contingency drives then exit out of the recovery process by entering *CANCEL*. Proceed to the RECOVERY COMPLETE CANCEL BASED section.
RECOVERY COMPLETE FULL	If recovery is not required then issue the command *ALTER DATABASE OPEN;* at the SQLDBA> prompt. Exit out of the SQL*DBA and **Redo Log Restore** program. The ownership of the restored data (.dbf) files should be restored from owner <*oradba*> to owner <*oracle*>. To do this using the current software, follow the following procedure: From the <*oradba*> account, enter the command *sqlplus / @/h/COTS/RDBMS/scripts/chmod_dbf.sql* After completion of this script, prepare the generated file for execution by entering *chmod 777 /tmp/dbfs.csh* Next, login to the unix <*root*> account and enter the command */tmp/dbfs.csh* After completion of the script, delete the generated file by entering the command *rm /tmp/dbfs.csh* Logout of the unix <*root*> account.

RECOVERY SEQUENCE	PROCEDURE
RECOVERY COMPLETE CANCEL BASED	Once the *CANCEL* option is used in the recovery command the database can be opened with the *ALTER DATABASE OPEN RESETLOGS;* at the SQLDBA> prompt. Exit out of the SQL*DBA and **Redo Log Restore** program. The ownership of the restored data (.dbf) files should be restored from owner <*oradba*> to owner <*oracle*>. To do this using the current software, follow the following procedure: From the <*oradba*> account, enter the command *sqlplus / @/h/COTS/RDBMS/scripts/chmod_dbf.sql* After completion of this script, prepare the generated file for execution by entering *chmod 777 /tmp/dbfs.csh* Next, login to the unix <*root*> account and enter the command */tmp/dbfs.csh* After completion of the script, delete the generated file by entering the command *rm /tmp/dbfs.csh* Logout of the unix <*root*> account.

4.7 RESTORE SUPPLEMENTARY USER BACKUP

If supplementary user backups were executed recently (refer to Paragraph C.2.10), you have the option to restore user permissions tables, Oracle user accounts, roles, grants and synonyms independently from a complete database recovery. Restoration of supplementary user backups is not necessary, and may result in loss of recently entered data if you are restoring the entire database using a full, cumulative or redo log backup.

Supplementary user backups are stored under the path **/h/SMDB/Scripts/SM_bld_tables**. If you have previously moved the backup files to tape (or renamed the files), move them back to the path **/h/SMDB/Scripts/SM_bld_tables**. Ensure that the file names exactly match the generated file names shown in Table C-5. Carefully check the date and contents of each script.

Login as the Unix user <*oradba*> from the GCCS globe. Enter <*cd /h/SMDB/Scripts/SM_bld_tables*> to change directories to the database scripts area. Select the restoration script(s) you wish to execute from Table C-10. You may execute one or all of the scripts, in sequence, from top to bottom. Keep in mind that unless all restore scripts shown are run successfully, GCCS Database user data may be out of sync. Remember that supplementary user backups are point in time backups that will remove existing data and restore user data exactly as it appeared at the time of the last backup.

Table C-10: Restore Supplementary User Backup.

Restore Command	Description	Results File
rstr_user_perm.sh	Truncate table and import all rows for the following user permissions tables: **JOPES_USER**, **USER_FUNCTION_PERMISSION**, **OPLAN_SERIES_PERMISSION**, **USER_OPLAN_PERMISSION**, **OPLAN_ACCESS**.	*rstr_user_perm.log*
rstr_users.sh	Drop existing externally identified GCCS Oracle user accounts, and restore user accounts from backup.	*rstr_users.lis*
rstr_roles.sh	Drop existing GCCS Oracle roles, and restore roles from backup.	*rstr_roles.lis*
rstr_grants.sh	Drop existing GCCS Oracle grants, and restore grants from backup	*rstr_grants.lis*

After completion of the restore scripts, check the results file indicated in Figure C-5 for errors. Correct database object dependencies and re-execute each script, if required.

At this time there is no .sh file used to restore synonyms. To restore synonyms, enter <*sqlplus -s oradba @rstr_synonyms.sql*>. Enter the <*oradba*> password if needed. After completion of the rstr_synonyms.sql script you must exit sqlplus. To exit sqlplus enter <*exit*>. Check the results file **/h/SMDB/Scripts/SM_bld_tables/rstr_synonyms.lis** for errors. Correct database object dependencies and re-execute the script, if required

SECTION C.5 — DATABASE RECOVERY

C.5.1 OVERVIEW

Database recovery encompasses numerous failure conditions. Table C-11 summarizes failure conditions that may require database recovery. As indicated in the figure, ORACLE handles some failures with minimal site administrator intervention. The scope of this document is primarily human error, media failure, and disaster.

The potential for human error during backup and recovery may be reduced by adopting standard site procedures. Procedures for tape organization, labeling, and the backup binder are described in Section C.3, Backup Library Management. Additional procedures are provided in the TDBM handbook and suggested in ORACLE and Unix system administration documentation.

Media failure includes numerous error conditions and consumes the most recovery resources. Operating system and database knowledge is required to diagnose and perform recovery procedures. Section C.6, Administrator Qualification, describes the required skills in detail. During media recovery, restore scripts must be used with associated ORACLE procedures; Paragraph C.5.2, Media Failure, describes these relationships .

Disaster situations involve loss of data or control files WITH loss of backup tapes required to perform local recovery. Under these conditions, you must make a careful inventory of damage to determine if recovery from another C/S is required. Paragraph C.5.3, Disaster, provides a discussion of Disaster Recovery.

Table C-11: Database Failure Conditions.

FAILURE CONDITION	DESCRIPTION	EXAMPLES	WHAT CAN ORACLE DO?
USER ERROR	- Database failures attributed to human error. - Caused by end users or by functional and technical database administrators.	- An end user accidentally deletes a CARRIER data record - An administrator deletes the CARRIER_ITINERARY ORACLE table structure.	Administrators can do little to prevent user errors, however: - Increased training on database and application principles will reduce mistakes. - ORACLE security profiles protect against excessive data loss.
RDBMS SQL STATEMENT FAILURE	A logical failure in the handling of a SQL statement within the ORACLE RDBMS.	- A CARRIER_MANIFEST record cannot be added because there is no physical space available.	- ORACLE automatically rolls back any effects of the statement and returns control to the user or user program. - The user should contact the DBA/SA to correct the space problem.
RDBMS PROCESS FAILURE	A failure in an ORACLE RDBMS user, server, or background process of a database instance.	- The GCCS application software terminates unexpectedly. - The server crashes while sending a network transaction.	- If the process is a user or server process, ORACLE automatically rolls back the current transaction and releases used resources. - If the process is a background process, the database must be shut down and restarted. Database integrity is restored as part of the restart process.
NETWORK FAILURE	A failure of network communication lines or software.	The LAN goes down on a SQL*NET C/S.	- ORACLE resolves the error as a process failure.
DATABASE INSTANCE FAILURE	A hardware or software problem that shuts down an ORACLE database instance (System Global Area and background processes).	- The C/S power supply goes down. - The Unix OS crashes unexpectedly.	- Recovery is automatically performed as part of the next instance startup; the database is restarted and rolled back to a consistent state prior to failure.
MEDIA FAILURE	Magnetic media damage to online redo log files, control files, and data files.	- A disk head crashes. - A disk develops bad tracks. - A disk controller goes bad.	ORACLE can guarantee full media recovery if it uses: - Mirrored control files AND - Mirrored, archived online redo logs AND - Operating system image backups of data files.

FAILURE CONDITION	DESCRIPTION	EXAMPLES	WHAT CAN ORACLE DO?
DISASTER	Magnetic media damage to control files or data files AND loss of backup tapes.	- Acts of God. - Any situation involving destruction of disk drives AND destruction of required backup tapes.	Every disaster situation is unique. General disaster recovery procedures are provided in Paragraph C.5.3, Disaster.

C.5.2 MEDIA FAILURE

Recovery from media failure requires consistent application of backup procedures and knowledge of ORACLE and Unix operations and commands. The section "Recovering a Database" in the *ORACLE7 Server Administrator's Guide* provides detailed procedures for media recovery.

ORACLE error messages may indicate the need for media recovery (refer to the ORACLE and Server Messages and Codes Manual). Table C-12 illustrates common media recovery error messages.

Table C-12: ORACLE Media Recovery Error Messages

ORACLE ERROR	MESSAGE
ORA-00204	**"error in reading control file 'name' (block num, # blocks num)"**
ORA-01113	**"file name needs recovery"**
ORA-01168	**"file name: bad physical block size of num bytes expecting num"**
ORA-01178	**"file name created before last CREATE CONTROLFILE, cannot recreate"**

.

After receiving such an error message, you should determine files that require recovery. **V$RECOVER_FILE** may be used when the database is offline; a typical query is provided below:

```
SELECT FILE#, ONLINE, ERROR, CHANGE#, TIME
   FROM V$RECOVER_FILE;
```

After determining file numbers that require recovery, the file name may be determined using the **V$DATA_FILE** view:

```
SELECT NAME, FILE#, STATUS
   FROM V$DATAFILE
   WHERE FILE# = "FILE#";
```

After obtaining the list of file names that require recovery, you must decide which media recovery action(s) are required.

Table C-13 provides guidance for restore procedures to fix common media failures.

To use the table, follow the instructions below:

1. Use the five columns on the left of the table to determine the types of files that are damaged: data files, online redo logs, archived redo logs, or the control file. Find the row that matches your media failure condition.

2. Read across the row to the center column "ORACLE recovery required" for a general description of the most probable ORACLE action required. (Refer to ORACLE documentation for specific details of each recommended action.)

3. Read farther right across the row to the four restore procedures available through the Backup and Recovery menu. The sequence numbers, read from left to right, indicate the sequence in which each of the restore procedures should be applied. Use the sequence with the least number of steps; this determination is dictated by the extent and timing of media damage.

4. If you choose an advanced recovery procedure (i.e. incomplete media recovery, recovering only one tablespace) to save time, advanced knowledge of Unix tape commands and Oracle is required. For restoration of individual files from tape without using menu options, use the *tar* command with a blocking factor (*-b*) of 112.

Table C-13: Restore Procedures: Common Media Failures.

Number of types damaged	Data file	Online redo log file	Archived redo log file	Control file	ORACLE recovery action required	Restore full backup	Restore cumulative backup	Restore redo log backup
1	X				Use complete media recovery	1	1 2	1 2 3
		X			Recover the missing files **(if already archived, may copy from backup disk)**	NO	NO	(if archived to tape)
			X		Take a new backup of all data files: **Perform full backup**	NO	NO	NO
				X	Recover the missing file; **use the mirrored file if possible**	(control file only)	NO	NO
2	X	X			Use incomplete media recovery **(to the last recoverable log)**	1	1 2	1 2 3
	X		X		Use incomplete media recovery **(to the last recoverable log). Perform full backup**	1	1 2	1 2 3
	X			X	Recover the control file **(use the mirrored file if possible)**, also recovering data files	1 (w/ control file)	2	3
		X	X		Use incomplete media recovery **(to the last recoverable log)**	NO	NO	(if sequence restorable)
		X		X	Recover the control file **(use mirrored copy if possible)**, resetting the redo log **(use mirrored copy or backup disk archive if possible)**	(control file only)	NO	(if archived to tape)
			X	X	Recover the control file (use mirrored copy if possible), using incomplete media recovery **(to the last recoverable log)**. Take a new backup of all data files: **Perform full backup**	(control file only)	NO	NO

Number of types damaged	Data file	Online redo log file	Archived redo log file	Control file	ORACLE recovery action required	Restore full backup	Restore cumulative backup	Restore redo log backup
3	X	X	X		Use incomplete media recovery **(to the last recoverable log)**	1	1 2	1 2 3
	X	X		X	Recover the control file **(use mirrored copy if possible)**, also recovering other files **(to the last recoverable log)**	1	2	3 (if sequence restorable)
	X		X	X	Recover the control file **(use mirrored copy if possible)**, also recovering data files **(to the last recoverable log)**. Perform a full backup if a break exists in redo log sequence**.**	(control file) 1	1 2	1 2 3
		X	X	X	Recover the control file **(use mirrored copy if possible)**, resetting the redo log. Perform a full backup.	(control file)	NO	NO
4	X	X	X	X	Recover the control files, also recovering other files. Perform a full backup. **Use disaster recovery if required (refer to ParagraphC. 5.3, DISASTER)**.	1	2	3

C.5.3 DISASTER

Disaster recovery includes a wide range of possible scenarios; however, a disaster recovery should not be a common occurrence. Volume Manager’s disk mirroring is the primary recovery mechanism. The secondary mechanism is local recovery from tape backups. Any situation involving loss of data or control files, and loss of backup tapes required to perform the local recovery, is considered a disaster. Under these conditions, the user must make a careful inventory of damage to determine if recovery from another C/S is required.

In most recovery situations, backup tapes are not damaged. In these situations the user will follow the procedures described in Paragraph C.5.2, Media Recovery.

If disaster recovery is required and at least one other C/S with similar functional data is operational, the user is to follow the general disaster recovery procedure described in Table C-15.

Every disaster situation is unique, and some situations may require advanced ORACLE7 knowledge. Contact the network expert site if further assistance is required.

The disaster recovery procedures provided in this section do not specifically address DoD and local information security considerations.

Table C-14: Legend.

LEGEND:

P - Providing Oracle GCCS Database site
R - Receiving Oracle GCCS Database site
S - Selected Oracle GCCS Database sites that share P & R’s data

<*Italics*> - Italicized within brackets denotes command line input instructions, to be followed by execution of the command by pressing the ENTER key. Brackets are omitted where the symbol ‘>’ is used as part of the command text.

UPPER - Standard upper case denotes menu choices.

Table C-15: Database Disaster Recovery.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
***	Assumption 1	P R	All Oracle user accounts from the providing site (P), will be recreated at the receiving site (R) during Step 11. If supplementary user backups were performed at R, Oracle user accounts may be restored from tape; this option removes all Oracle accounts previously restored from P.	The WASO should ensure that parallel Oracle accounts are maintained at sister sites. This eliminates the potential for error during supplementary user restoration.
***	Assumption 2	P R	The server domain entry for <sm> (both USERID and password) must be recreated at R exactly as they appear at P. This is done through the TDS Network Management screen in System Services. The user must belong to the JADMIN group.	If R does not contain the same <sm> entries, the connect scripts will need to be updated at every other site. Do this by deleting the connect script with the recovered site in the name. For example, if ARPAC was the recovered site, delete the connect script, "hawaii_connect_script." Someone with JADMIN permissions should type the following. *<cd /h/SM/Scripts/tds>*, *<ls -l>* to verify script name, then *<rm hawaii_connect_script>*. The system will regenerate this script when transactions flow to or from the restored site.
***	Assumption 3	P R	The Oracle import utility (Step 11) will attempt to recreate all data files from P. P must contain all the database segments required by R. If P has database segments that were not previously installed on R, the Oracle import utility will generate multiple non-fatal errors. These errors will make the R_gccs_full.log import results file difficult to read.	Recovery from a sister site with exactly the same database segments reduces the potential for human error during recovery.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
***	Assumption 4	P R	Users at R have saved local OPLANs to tape or disk.	With no backup tapes or disk files, local OPLANs will be lost.
***	Assumption 5	P R	P will contain all OPLANs distributed to R.	
***	Assumption 6	P R	Both P and R will identify a filesystem with read/write access to the Unix <oradba> user, with space available for the full complete export. Presently, 1,000,000 KB of available space <*df -a*> is required. If the filesystem /oracle/smback is selected, the space used by large exports will cause unpredictable performance of redo log backup/recovery software. If the filesystem /tmp is selected, export data may be lost if the server is shut down or crashes. Throughout this document, the filesystem selected as a disaster recovery staging area will be referred to as avail_filesystem.	
***	Assumption 7	P R	A Unix ftp transfer user exists, with access to the directory avail_filesystem, and the ability to transfer files between P and R.	

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
1	Choose P	P	Identify the P database that most exactly matches the OPLAN data and installed database segments previously installed on the R. When making this determination, the best P contains exactly the same installed database segments as R. P is acceptable if it contains more database segments than R, and unacceptable if it contains fewer installed database segments than R.	R will be recovered from another existing and functional Oracle database server. During the Oracle import, database files will be recreated in data (.dbf) directories. Because many data directories are created during segment installation, extra data files from P will fail on R; this will cause non-fatal errors on subsequent imported database objects that depend on the existence of failed data files (tablespaces).
2	Disable Transaction Processing and Distribution at R	R	Turn off transaction processing by choosing SYSTEM SERVICES UTILITY under main menu, then TRANSACTION PROCESSING MANAGEMENT, and finally TERMINATE. Turn off TRANSACTION DISTRIBUTION by disabling it under the TRANSACTION DISTRIBUTION MANAGEMENT menu.	R should not propagate database changes to other sites during recovery.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
3	Prepare R for Recovery	R	Take all necessary steps to bring R back up as a functional server, with the GCCS software installed (with no database or data). For example, all hardware problems must be fixed and the operating system must be up. When installing Oracle, only the software should be installed. A GCCS Database instance will be created later through scripts in Step 7. Install the GCCS application software. However, since a GCCS Database instance does not exist at this point, the GCCS application software cannot be run or tested. Take note of any application that requires existence of the database, and install these applications after the database installation.	Prepare hardware, operating system and applications for database recovery. Reinstall generic Oracle database software.
4	Bring All Sites Sharing R Data to Zero Flow	P S	Steps 4a and 4b are necessary to ensure database synchronization. In some situations, it may be advantageous to perform disaster recovery without bringing all sites to zero flow. To do this, skip to Step 4c. In this less desirable scenario, the user must remember to perform an additional synchronization cross-check after recovery, as some database synchronization problems are likely (refer to Step 26).	Database synchronization after disaster recovery is best ensured by bringing all sites sharing R's data to zero flow.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
4a	Process Send Queue Backlogged Transactions	P S	Perform the following steps at all sites sharing R's data: • Launch the GCCS System Services Application. • Under Menu choose MONITORS and then choose SEND QUEUE DETAIL. • When QUEUED column values are zeros at all sites sharing R's data, then all transactions have been sent. Proceed after all transactions are processed. • Turn off TRANSACTION DISTRIBUTION by disabling it under the TRANSACTION DISTRIBUTION MANAGEMENT menu at all sites sharing R's data.	For each site that has P & R's data, stop tds processing the Send Queue. The tdc_ready directory should be monitored until empty. Ensures that all transactions that originated on the server have been sent. After all transactions are processed, transaction processing and distribution are disabled.
4b	Process Receive Queue Backlogged Transactions	P S	At P, monitor tds_ready and Receive Queue until all transactions are processed. • Under Menu choose MONITORS and then choose RECEIVE QUEUE DETAIL. • When the TOTAL QUEUED column value contains zero, then all transactions have been processed. Proceed after all transactions are processed. • Turn off transaction processing by choosing SYSTEM SERVICES UTILITY under main menu, then TRANSACTION PROCESSING MANAGEMENT, and finally TERMINATE. Repeat this step at all sites sharing R's data.	Ensures that all transactions being sent to the site have been received. After all transactions are processed, transaction processing is disabled.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
4c	Less Desirable Alternative to Zero-Flow	P	Use this option only if you have authorization to skip Steps 4a and 4b. (Refer to instructions, Step 4). At P only, turn off TRANSACTION DISTRIBUTION by disabling it under the TRANSACTION DISTRIBUTION MANAGEMENT menu. This must be performed by the same user that started it (this should be the <SM> Unix account). At P only, monitor tds_ready and Receive Queue until all transactions are processed. • Under Menu choose MONITORS and then choose RECEIVE QUEUE DETAIL. • When the TOTAL QUEUED column value contains zero, then all transactions have been processed. Proceed after all transactions are processed. At P only, turn off transaction processing by choosing SYSTEM SERVICES UTILITY under main menu, then TRANSACTION PROCESSING MANAGEMENT, and finally TERMINATE.	In some situations, it may be advantageous to perform disaster recovery without bringing all sites to zero flow. In this less desirable scenario, remember to perform an additional synchronization cross-check after recovery, as some database synchronization problems are likely (refer to Step 26).

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
5	Disable Automatic Backups During Recovery	P R	Disable automatic full, cumulative, and redo log backups at both P and R. Before disabling backups, perform a manual redo log backup (*Appendix C, System Services Administrator's Manual)*, to provide space for archived redo logs to accumulate. To disable backups, login as the Unix <oradba> user from the GCCS globe. Enter the command <*br_main*> to open the backup and recovery main menu. Select <A> for the Automatic Backup Menu. Make a note of the current backup settings, so that you can reset the same times later (Step 8a for P, and Step 29 for R). Disable each backup by entering <DF>, <DC>, and then <DR>. Exit to the Unix prompt.	Disable automatic backups to reduce distractions.
6	Notify Users at Providing Site	P	Ensure that all Oracle GCCS Database users are logged out now, and remain logged out for the duration of Step 6. One way to prevent user activity is to enable a restricted session. Only use this method during Step 6b, as user accounts specified in Step 6a cannot access the database in restricted mode.	To ensure database consistency during the export and row counts, users should not be permitted to access the database at P.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
6a	Perform Table Counts at P	P	Immediately before execution of the export (see Step 6b), run a row count script at P, for use later in Step 11. Throughout this document, the filesystem selected as a disaster recovery staging area will be referred to as “avail_filesystem” (see Assumption 6). As the Unix user <oradba>: *<cd /h/SMDB/Scripts/SM_bld_tables>* *<sqlplus table_master>* Enter the table_master password. *<@count>* *<mv count_tbl.lis avail_filesystem/P_count_table_master.lis >* *<sqlplus gsorts>* Enter the gsorts password. *<@count>* *<mv count_tbl.lis avail_filesystem/P_count_gsorts.lis >* *<sqlplus npgdba>* Enter the npgdba password. *<@count>* *<mv count_tbl.lis avail_filesystem/P_count_npgdba.lis >* *<sqlplus mepesdba>* Enter the mepesdba password. *<@count>* *<mv count_tbl.lis avail_filesystem/P_count_mepesdba.lis >* Enter the following command: *ls /h/OJEPES/usernames > avail_filesystem/P_JEPES_users.lis* Enter the following command: *ls /h/OLSAFE/usernames > avail_filesystem/P_LOGSAFE_users.lis*	Table counts will be ftp’d to the receiving site along with the export file, to help in determining if the import was successful at the receiving site (Step 11b). List the LOGSAFE and JEPES users, for use in Step 18c.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
6b	Export Full Database from P	P	As the Unix <oradba> user, create a parameter file, avail_filesystem/exp_full_db.par, which contains the following lines: *LOG=P_gccs_full.log* *FILE=P_gccs_full.dmp* *FULL=Y* *COMPRESS=N* *INDEXES=Y* *STATISTICS=NONE* Using the export parameter file, enter the following command at the Unix prompt, as the Unix <oradba> user: *exp / parfile=avail_filesystem/exp_full_db.par* If you are prompted for a username/password, enter the username <*oradba*>, then oradba's <*password*>. Step 8 may be performed immediately after the export is complete.	Exports all the data on P into a dump file. The export will likely require two hours or longer. The ftp will likely require five hours or longer. One way to prevent user activity is to enable a restricted session. Only use this option during Step 6b. As the Unix user <oradba>: *<sqldba lmode=y>* *<connect internal>* *<alter system enable restricted session;>* *<exit>* Do not forget to disable the restricted session at the end of Step 6b. From the Unix user <oradba>: *<sqldba lmode=y>* *<connect internal>* *<alter system disable restricted session>* *<exit>*

S T E P	STEP TITLE	S I T E	ACTIONS & COMMANDS	EXPLANATION
7a	Prepare to Recreate the GCCS Database Instance	R	Log in as the Unix user <oradba>. Enter: <cd $ORACLE_HOME/dbs > <sqldba lmode=y> <connect internal> <spool datafile.lis> <select name from v$datafile order by name;> <shutdown abort> <exit> From <oradba>'s Unix prompt, remove (rm) all files shown in the results file $ORACLE_HOME/dbs/datafile.lis. Be aware that a few segments allow installer selection of data file physical location; this could lead to tablespace creation errors during the import, if datafiles on P are written to filesystems that are too small, or do not exist on R. If this occurs, call the expert site for assistance in manually creating the tablespace in a different location, and reimporting tablespace data. Remove the Oracle control files and system files: <rm $ORACLE_HOME/dbs/*GCCS*.ctl> <rm /home10/oracle/*GCCS*.ctl> <rm /oracle/smback/arch/*GCCS*.ctl> <rm $ORACLE_HOME/dbs/*GCCS*.dbf> Edit the file crdbGCCS.sql, adding a comma after the 'M' on the first datafile specification line. (Omission of the comma prevents accidental re-creation of the database.)	Creating a new GCCS instance, as described in the following steps, recreates any remaining GCCS instance database objects. **Note:** It is not unusual to discover data files located in oracle directories, but not recognized by the GCCS database (GCCS recognizes only the files in datafile.lis). This occurs if a tablespace is dropped without a subsequent step to remove data files, or if an Oracle database with a SID other than GCCS owns the data files. Use caution, and contact the expert site to determine if data files not listed in datafile.lis can be removed.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
7b	Create Database	R	<cd $ORACLE_HOME/dbs> Run create database, Part I: *<sqldba lmode=y>* (Do not connect internal) *<@crdbGCCS>* *<exit>* Carefully review the file crdbGCCS.lst for errors. To correct errors, shutdown the database, delete any files created in the script, and re-execute crdbGCCS. Only after the above steps completed without errors,run create database, Part II: *<sqldba lmode=y>* **Note:** (Do not connect internal) *<@crdb2GCCS>* Make a note and evaluate any errors relating to tablespace creation, appearing on the screen during the first several minutes of script execution. After tablespace creation, Oracle software product utilities are called: errors in these scripts are not unusual (often caused by existence or non-existence of database objects), and may be ignored. *<exit>*	The script crdbGCCS.sql will create the GCCS Database instance. The script crdb2GCCS.sql will create rollback segments, tools tablespaces, reload the database dictionary, etc. These steps create a generic GCCS Database without any user data files.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
7c	Modify System Password and Create <oradba> User	R	Login into sqldba from the <oradba> Unix account: *<sqldba lmode=y>* *<connect internal>* Modify the system user account password by specifying the new password: *<alter user system identified by new password*;*>* *<exit>* Create the user <oradba>, identified externally (i.e., after <oradba> logs into the operating system, <oradba> can access sqlplus / without a password). From the Unix user <oradba>: *<sqlplus system> (Enter system password)* *<@/h/COTS/RDBMS/scripts/create_user oradba>* *<sqlplus system> (Enter system password)* *<grant dba to oradba;>* *<exit>*	Modify the <system> password by specifying your password instead of new_password in the procedures to the left. Create the <oradba> user account with the dba role. <oradba> is identified externally during disaster recovery to allow backwards software compatibility. Check with the Database expert site during cleanup of user accounts (Step 23) to determine if the <oradba> account should be identified with a password after recovery.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
7d	Verify that the Database is not Archiving Redo Logs	R	Verify that the database is in NOARCHIVELOG mode. To do this, log into sqlplus as the Unix user <oradba>: *<sqlplus />* *<select * from v$database;>* The column LOG_MODE should contain the value NOARCHIVELOG. If set correctly, skip to Step 7e. If LOG_MODE contains the value ARCHIVELOG, disable redo log archiving: Log in as the <oradba> user. From the Unix prompt: *<sqldba lmode=y>* *<connect internal>* *<shutdown normal>* If the database does not come down within several minutes, open a new window, and repeat the steps above, using <*shutdown abort*> instead of the command <*shutdown normal*>. If <*shutdown abort*> is used, also enter the commands <*startup open*> and <*shutdown normal*> before continuing. *<startup mount restrict;>* *<alter database noarchivelog;>* *<alter database open;>* *<alter system disable restricted session;>* *<exit>*	Archiving should be disabled until all data in the database is recovered.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
7e	Rebuild Database Structures not yet in Baseline	R	**Note:** Check with the database expert site or OSF for the Integration Contractor's rebuild instructions before proceeding with Step 7e; there may be additional or updated instructions. One or more of the scripts executed in Step 7e may be required, depending on the /h/ORA* baseline at the time of recovery. As the unix user <oradba>: *<sqlplus />* *<@/h/SMDB/Scripts/SM_bld_tables/expand_sm_tbsp>* This script will return numerous error messages; it is run primarily to expand selected database creation objects. *<exit>* *<sqldba lmode=y>* *<connect internal>* *<create rollback segment r0 tablespace system storage (initial 16k next 16k minextents 2 maxextents 20);>* *<alter rollback segment r0 online;>* *<@/h/COTS/RDBMS/scripts/alter_db_P4>* *<@/h/COTS/RDBMS/scripts/alter_db_P6>* *<exit>* Review the files /tmp/alter_db_P4.log and /tmp/alter_rbs.lst for errors.	It is possible that some database structures will need to be updated, if recent structural changes are not yet reflected in the create database scripts. These changes involve critical database objects such as rollback segments, online redo logs, and temporary space. Failure to perform this step correctly will most likely result in time-consuming failures during the database import. **Note 1**: You may notice a message such as define_editor = vi, or other unusual message during sqlplus logins. Disregard these messages and proceed. **Note 2:** If you receive the error "product user profile information not loaded", log into sqlplus as <system>, and execute the command *<@$ORACLE_HOME/sqlplus/admin/pupbld>*, then *<exit>*.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
7e	Rebuild Database Structures not yet in Baseline (continued)	R	As the unix user <oradba>: *</h/COTS/RDBMS/scripts/alter_ts.csh>* Review the file /tmp/alter_temp.log for errors. As the unix user <oradba>: *<sqldba lmode=y>* *<connect internal>* *<alter rollback segment r0 offline;>* *<drop rollback segment r0;>* *<@/h/COTS/RDBMS/scripts/alter_db_P6_disable>* *<exit>* Next, follow the cleanup steps below, derived from the /h/ORAP4/SegDescrip/PostInstall. Be especially careful when deleting data (.dbf) files. Execute the following commands as the unix user <oradba>: *<rm $ORACLE_HOME/dbs/log*1GCCS.dbf> *<rm $ORACLE_HOME/dbs/log*2GCCS.dbf> *<rm $ORACLE_HOME/dbs/log*3GCCS.dbf>	(Continued)
8a	Reenable Oracle Database Automatic Backups	P	At P only, reenable automatic full, cumulative, and redo log backups. Login as the Unix <oradba> user from the GCCS globe. Enter the command <*br_main*> to open the backup and recovery main menu. Select <A> for the Automatic Backup Menu, then enable backups by selecting <F>, <C>, and <R>. Exit to the Unix prompt.	Backups were previously disabled; reenable automatic backups tailored to the site schedule.

S T E P	STEP TITLE	S I T E	ACTIONS & COMMANDS	EXPLANATION
8b	Move Export and Table Count Files from P to R.	P R	Move the export file P_gccs_full.dmp as well as the table count files to R into the directory avail_filesystem. Before moving the files, ensure that the P_gccs_full.log file indicated successful completion of the export. The <oradba> user at P, should enter the following commands: *<cd avail_filesystem>* *<chmod 777 P_* >* Log into Unix at R, as a ftp transfer Unix user for which you have the password at P. Enter the following commands: *<cd avail_filesystem>* *<ftp providing_server_name>* Enter password for the selected ftp transfer Unix user at the providing site. From the ftp prompt: *<cd avail_filesystem>* (This is the avail_filesystem for P.) *<bin>* *<mget *.lis>* Enter <*Y*> for each file to be transferred. *<get P_gccs_full.dmp* When complete, enter <*bye*>.	Gets the data in P’s database onto R. This file will be used in the next step for importing into R’s database. The ftp may take five hours or more, depending on network transmission rate.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
9	Ensure that the Database is not Archiving Redo Logs	R	Verify that the database is in NOARCHIVELOG mode. To do this, log into sqlplus as the Unix user <oradba>: *<sqlplus />* *<select * from v$database;>* *<exit>* The column LOG_MODE should contain the value NOARCHIVELOG. If LOG_MODE is set correctly, skip to Step 10a. If the LOG_MODE value contains ARCHIVELOG, disable archiving as described in Step 7d.	During recovery, archiving serves little purpose and results in accumulation of archived redo logs which can fill /oracle/smback/arch to capacity. Archiving will be turned on again after recovery is complete. **Note:** The database will be in archivelog mode only if the database has been shutdown and reopened.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
10a	Take Small Rollback Segments Offline	R	Take all rollback segments except rb_batch offline, to ensure a clean import. Log into sqlplus as the Unix user <oradba>: *<sqlplus />* *<alter rollback segment r01 offline;>* *<alter rollback segment r02 offline;>* *...repeat for rollback segments r03 through r11...* *<alter rollback segment r12 offline;>* Confirm the results of your commands by entering: *<select segment_name,status from sys.dba_rollback_segs;>* The results of this query should indicate that only the SEGMENT's SYSTEM and RB_BATCH are ONLINE.	During the import, all rollback activity should occur in the large batch-oriented rollback segment, RB_BATCH. Take all other rollback segments, except the mandatory SYSTEM segment, offline. If other segments are left online, large objects may fail on rollback space errors during the import.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
10b	Check the TEMP Tablespace Extent Settings	R	Still in sqlplus, check the NEXT EXTENT setting for the TEMP tablespace by entering: *<select tablespace_name,next_extent* *from sys.dba_tablespaces* *where tablespace_name = 'TEMP';>* *<exit>* The column NEXT_EXTENT should contain a value equal to or larger than 2408448 (BYTES). If set correctly, skip to Step 11. If NEXT_EXTENT contains a value less than 2408448 (BYTES), adjust the tablespace storage parameters: *<sqlplus />* *<alter tablespace TEMP default storage* *(NEXT 3M);>* *<exit>*	The TEMP tablespace has been adjusted several times to reduce fragmentation and allow optimal performance. If the next extent setting is below two Megabytes during the Import, objects may fail creation.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
11a	Import	R	From the <oradba> Unix prompt: *<cd avail_filesystem>* Create an import parameter file, imp_full_db.par, containing the following lines: *FILE=P_gccs_full.dmp* *LOG=R_gccs_full.log* *FULL=Y* *IGNORE=Y* *INDEXES=Y* *DESTROY=Y* Import using the following command as the Unix <oradba> user from the Unix prompt: *<imp / parfile=imp_full_db.par>*	Builds all of the tablespace structures, and creates data, from P's database into R's database. For a rough estimate of import time: Best Case: One hour for each 40MB of the compressed dump (.dmp) file Worst Case: One hour for each 20MB of the compressed dump (.dmp) file.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
11b	Import (Compare Row Counts)	R	Before reviewing the IMPORT log file (Step 11c), compare counts of the data rows at P versus the rows imported at R. Do not open any applications before running count.sql at R. If the counts are the same, the IMPORT was successful (keep this in mind when proceeding to Step 11c). As <oradba>: *<sqlplus / >* *<alter user GSORTS identified by new_password;>* *<alter user TABLE_MASTER identified by new_password;>* *<alter user MEPESDBA identified by new_password;>* *<alter user NPGDBA identified by new_password;>* *<alter user SYSTEM identified by new_password;>* *<exit>* *<cd /h/SMDB/Scripts/SM_bld_tables>* *<sqlplus table_master>* (Enter the table_master password) *<@count>* *<mv count_tbl.lis avail_filesystem/R_count_table_master.lis >* *<sqlplus gsorts>* (Enter the gsorts password) *<@count>* *<mv count_tbl.lis avail_filesystem/R_count_gsorts.lis >* *<sqlplus npgdba>* (Enter the npgdba password) *<@count>* *<mv count_tbl.lis avail_filesystem/R_count_npgdba.lis >* *<sqlplus mepesdba>* (Enter the mepesdba password) *<@count>* *<mv count_tbl.lis avail_filesystem/R_count_mepesdba.lis >*	Data row counts provide a good indication if the import was successful. Some applications, such as RDA, alter the contents of temporary tables during normal operation. The row counts should not be considered reliable if: • Any applications or Oracle users were active at P during the activities described in Step 6, or, • Any applications or Oracle users were active at R during the activities described in Step 11.

<table>
<tr><th>S
T
E
P</th><th>STEP TITLE</th><th>S
I
T
E</th><th>ACTIONS & COMMANDS</th><th>EXPLANATION</th></tr>
<tr><td><b>11b</b></td><td>Import (Compare Row Counts)</td><td>R</td><td>Next, compare P’s row counts with R’s row counts using the Unix diff command, as the Unix <oradba> user. Enter the following four commands:
<i>sdiff P_count_table_master.lis R_count_table_master.lis >pr_diffs_table_master.lis</i>
<i>sdiff P_count_gsorts.lis R_count_gsorts.lis>pr_diffs_gsorts.lis</i>
<i>sdiff P_count_npgdba.lis R_count_npgdba.lis >pr_diffs_npgdba.lis</i>
<i>sdiff P_count_mepesdba.lis R_count_mepesdba.lis >pr_diffs_mepesdba.lis</i>
Look for differences using the commands shown in the explanation column to the right. If all differences are explainable, the IMPORT was most likely successful. Review Step 11c, and skip to Step 12.</td><td>Use the following commands to interpret the diff results file. Keep in mind that if for any reason, one line is off between the providing site (P) .lis file and the receiving site (R) .lis file, the diffs file will contain entries.
For example:
<i>more pr_diffs_gsorts.lis</i> | <i>grep</i> ‘|’ returns
P value | Different R value, same line
<i>more pr_diffs_gsorts.lis</i> | <i>grep</i> ‘ >’ (make sure you include a blank space before the > symbol in the line above.) returns:
| Value in R file only
<i>more pr_diffs_gsorts.lis</i> | <i>grep</i> ‘<’ returns:
Value in P file only |
Entries in pr_diff_gsorts.lis with no "|",">", or "<" indicate equal values on different lines.</td></tr>
</table>

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
11c	Import (Review .log File)	R	Carefully review the R_gccs_full.log file for errors. Ignore messages referring to database segments that were not previously installed at R (refer to Assumptions, Item 1, at the beginning of this table). Pay attention to table or index failures. It is likely that some table or index create statements will fail due to lack of space (refer to Step 11). If P contains segments not installed on R, the R_gccs_full.log file will contain **MANY** cascading errors, and will be difficult to interpret. If a user account is associated with an error, it may be possible to login to P, and identify the data file used by the user. If the corresponding filesystem does not exist on R, the segment for that user may not be required on R. To execute this query, have the DBA at P login as the Unix user <oradba>. Have the DBA at P login to sqlplus, and execute a query to identify data files associated with a user (user CCJJ99 in this example): *<sqlplus />* (If this does not work, enter *<sqlplus oradba>*, then *<password>*.) *<select tablespace_name, file_name* *from sys.dba_data_files* *where tablespace_name in* *(select tablespace_name* *from sys.dba_tables* *where owner = ‘CCJJ99');>*	If the providing site contains segments not installed on the receiving site, the .log file will contain many cascading errors. This is usually caused by initial failure to create a tablespace because corresponding filesystems do not exist, followed by related errors when objects in the missing tablespace fail creation. In this case, expect a very large .log file with many thousands of errors. Here are some vi text editor commands you should know: h Move left l Move right j Move down k Move up $ End of line Ctrl U 200 Scroll up 200 lines Ctrl D 1000 Scroll down 1000 lines Ctrl F Forward screen Ctrl B Backward screen Shift G Go to End 1000 Shift G Go to line 1000 Esc /ORA- Search for error message “ORA-” Esc :q Quit Make sure Caps Lock is off. Enter <set term = vt100> at Unix prompt is display is jumbled.

S T E P	STEP TITLE	S I T E	ACTIONS & COMMANDS	EXPLANATION
12	Recreate Objects Failing Import	R	Skip to Step 13 unless the R_gccs_full.log (Step 11) indicates failure of mission-critical objects on R. In this case, reimport specified tables, and recreate indexes and constraints as described in the steps below. Be aware that a few segments allow installer selection of data file physical location; this could lead to tablespace creation errors during the import, if datafiles on P are written to filesystems that are too small, or do not exist on R. If this occurs, call the expert site for assistance in manually creating the tablespace in a different location, and reimporting tablespace data. Before calling the expert site, take the time to ensure that the tablespace is required on R.	Some space and fragmentation differences are possible between P and R; this step accommodates failures due to these problems.
12a	Add Space to Tablespace, (if required)	R	Add an additional data file to tablespaces containing objects that failed due to lack of space. For procedural instructions, refer to /h/SMDB/Scripts/SM_bld_tables/ DBA_readme.txt, EXPANDING A TABLESPACE.	R may lack space in certain tablespaces to accommodate all of P's data.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
12b	Reimport Tables That Failed During Step 11 Import.	R	Reimport tables that failed with creation errors during the full import. As the Unix user <oradba>: *<cd avail_filesystem>* Create (vi) an import parameter file, imp_table.par. Replace TABLE_OWNER with the name of the table owner (i.e., TABLE_MASTER). Replace TABLE1, TABLE2 with the actual table names you wish to reimport. Import tables from only one owner at a time: *FILE=P_gccs_full.dmp* *LOG=R_gccs_table.log* *FULL=N* *FROMUSER=TABLE_OWNER* *TOUSER=TABLE_OWNER* *TABLES=(TABLE1, TABLE2)* *INDEXES=Y* From the Unix user <oradba>: *<imp / parfile=imp_table.par>* Carefully review the R_gccs_table.log file for errors. If you note failures due to referential integrity constraints, drop referential integrity constraints (Step 12c), and try this step again.	A table mode import reads the full export .dmp file, and recreates only those tables specified in the import parfile. Tables should be imported with the parent table first, then the child, to avoid violating referential integrity constraints.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
12c	Drop Referential Integrity Constraints	R	Perform this step only if required (refer to 12b). At the Unix prompt from the <oradba> user: *<sqlplus table_master> (enter password)* *<@/h/SMDB/Scripts/SM_bld_tables/gen_drop_ref_cons.sql>* *<exit>* Reimport any tables that failed due to referential integrity constraints in Step 12b.	Remove referential integrity constraints to permit import of tables that failed to import in Step 12b.
12d	Recreate Objects Failing Import (if required)	R	At the Unix prompt: <cd /h/SMDB/Scripts/SM_bld_tables > <sqlplus table_master > (enter password) <@prm_key> <@sm_index> <@cr_index> <@import_ref> <@import_domn> <@rda_key_domn> <@import_range>	If there were errors with index or constraint creation during import, these scripts will recreate the missing objects. **Note:** Ensure that statistics are deleted from all tables (i.e., *analyze table x delete statistics*) for each table in the database if you encounter constraint creation errors that are not caused by missing data values. Call the database expert site for assistance in creating this script.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
13	Truncate Selected Tables	R	At the Unix command line from the <oradba> user account enter: *<sqlplus / >* Within SQLPLUS enter: *<TRUNCATE TABLE table_master.send_queue;>* *<TRUNCATE TABLE table_master.failed_transaction_dist_log;>* *<TRUNCATE TABLE table_master.file_transition;>* *<exit>*	This step deletes the providing server's site-unique tables to avoid transaction failures and database anomalies.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
14	Resequence Selected Tables	R	To resequence selected tables, first determine the existing next sequence value. From the Unix <oradba> user account: < *cd/h/SMDB/Scripts/SM_bld_tables* > < *sqlplus table_master* > (Enter table_master password.) Execute each of the following statements and write the value returned on the blank line provided (if a query returns no information, write "0"): SQL STATEMENT VALUE *<select max(crr_itn_id) from carrier_itinerary;>* ____ ____ *<select crr_itn_id_seq.nextval from dual;>* _________ *<select max(out_trans_id) from send_queue;>* _________ *<select out_trans_id_seq.nextval from dual;>* _________ *<select max(in_trans_id) from receive_queue;>* _________ *<select in_trans_id_seq.nextval from dual;>* ____ ____ *<select max(crr_rmk_id) from carrier_remark;>* ____ ____ *<select crr_rmk_id_seq.nextval from dual;>* ____ ____ Next, vi the /h/SMDB/Scripts/SM_bld_tables/cre_seq.sql file and enter the higher of the two numbers above rounded up (e.g., take the biggest number and add 1000 to it) in the "start with ..." entry for the specific area (example below is for the crr_itn_id_seq area in the script): create sequence crr_itn_id_seq start with *<enter the biggest number + 1000 here>* increment by 1 nocycle	This step ensures that the database internal value for each sequence number is set higher than sequence numbers already assigned to imported data. This prevents errors caused by duplication of primary key values.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
14	Resequence Selected Tables (continued)	R	Enter a new “start with” value for crr_itn_id_seq, out_trans_id_seq, in_trans_id_seq, and crr_rmk_id_seq. Next, do a *<:wq!>* to save the file. As the Unix user <oradba>: *<sqlplus table_master>* (Enter table_master password) *<@cre_seq>* When complete, ensure the new sequence numbers were updated successfully. From the Unix <oradba> user account: *< cd/h/SMDB/Scripts/SM_bld_tables >* *< sqlplus table_master >* (Enter table_master password.) *<select crr_itn_id_seq.nextval from dual;>* *<select out_trans_id_seq.nextval from dual;>* *<select in_trans_id_seq.nextval from dual;>* *<select crr_rmk_id_seq.nextval from dual;>* *<exit>*	(continued)

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
15	Update HOST_SITE Table	R	Update rows by entering the following commands in sqlplus. As Unix user <oradba>: *<sqlplus />* *<select * from HOST_SITE;>* *<delete from HOST_SITE where HOST_SITE_ID = ‘receiving site name’;>* *<delete from HOST_SITE where HOST_SITE_ID = ‘providing site name’;>* *<insert into HOST_SITE values (‘receiving site name’,’L’);>* *<insert into HOST_SITE values (‘providing site name’,’R’);>* *<select * from HOST_SITE;>* *<commit;>* *<exit>*	Update the HOST_SITE table to swap the providing and receiving site information. This step is to verify that the restored site now contains the letter “L” for local, and the sending site contains the letter “R” for remote.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
16	Enable TD and TP on R Site	R	Launch GCCS System Services at main menu. As the Unix user <SM>, choose TRANSACTION PROCESSING MANAGER, then ACTIVATE. The user can verify that the Transaction Processor is turned on by entering the following Unix command: *<ps -ef | grep tp>* If TP is active, the system will return the process /h/SM/progs/SM_graphic/tp_main. If TP is inactive, the system will return only the process "grep tp". If this occurs, use System Services to reinitiate the startup. Then enable TRANSACTION DISTRIBUTION under the TRANSACTION DISTRIBUTION MANAGEMENT menu. The user can verify that Transaction Distribution is turned on by entering the following Unix command: *<ps -ef | grep tds>* If TD is active, the system will return the following processes: /h/SM/progs/SM_tds/hcd_client -d /h/SM/app-defaults/.SSdefaults /h/SM/progs/SM_tds/xtds -d /h/SM/app-defaults/.SSdefaults /h/SM/progs/SM_tds/tdc -d /h/SM/app-defaults/.SSdefaults /h/SM/progs/SM_tds/tds -d /h/SM/app-defaults/.SSdefaults "/h/SM/progs/SM_tds/notify_tdc -d /h/SM/app-defaults/.SSdefaults If TD is inactive, the system will return only the process "grep tds". If this occurs, use System Services to reinitiate the startup.	Refer to the *System Services Administrator's Manual*.

S T E P	STEP TITLE	S I T E	ACTIONS & COMMANDS	EXPLANATION
17	Enable TD on P and Selected Sites	P S	Launch GCCS System Services at main menu. Enable TRANSACTION DISTRIBUTION under the TRANSACTION DISTRIBUTION MANAGEMENT menu.	Refer to the *System Services Administrator's Manual*.
18a	Restore Local Oracle User Accounts, If Desired, Step A.	R	Refer to Paragraph C.4.7, Restore Supplementary User Backups. First, check if a current user backup exists (<*ls -la*> the generated script /h/SMDB/Scripts/SM_bld_tables/ rstr_users.sql). Decide if you wish to restore this file (deleting all Oracle user accounts imported from the sister site). To estimate the time required to restore local users, first log into sqlplus as the Unix user <oradba>: *<sqlplus />* *<select count(*) from sys.dba_users;>* *<exit>* Multiply the count returned by one minute for each user. This is a rough estimate of the time required to drop user accounts imported from P. This estimate may vary markedly from site to site. Next, calculate the time required to restore grants to each user from the local backups. As the Unix user <oradba>: *<cd /h/SMDB/Scripts/SM_bld_tables>* *<grep -c "create user" rstr_users.sql>* Multiply the count returned by one minute for each user. This is a rough estimate of the time required to recreate user grants on R. This estimate may vary markedly from site to site.	Restoring users from backup is only recommended when P does not have similar Oracle user accounts and permissions as R. Oracle user accounts previously created on R during the import of P's database will be lost (refer to Assumption 1.) **Note:** If you restore from a local user backup, you will find that object grants are missing for users of some applications (those with tables not owned by GSORTS, TABLE_MASTER, MEPESDBA, or NPGDBA.) Recreate missing user grants by running application-specific user scripts after the database is restored. Performance for the drop user and restore grants steps will improve in the future, when all applications use Oracle ROLES.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
18b	Restore Local Oracle User Accounts, If Desired, Step B.	R	Decide if restoration of local user backups is worthwhile using the following criteria: 1. Estimated time required to restore users. 2. The currency of local user backups. 3. The consideration that only users identified externally (i.e. "sqlplus /" previously worked without a password) will be restored. Grants will only be restored to objects (i.e., tables, views) owned by GSORTS, TABLE_MASTER, MEPESDBA and NPGDBA. If you decide to keep user accounts already imported from P, and not to restore local user backups, skip to Step 19a. To proceed with restoration of local users, first take all small rollback segments offline (refer to Step 10).	The decision to restore local user backups will add considerable time to the recovery process. This time will be reduced in the future, when all applications user Oracle ROLES.

S T E P	**STEP TITLE**	**S I T E**	**ACTIONS & COMMANDS**	**EXPLANATION**
18c	Restore Local Oracle User Accounts, If Desired, Step C.	R	Drop JEPES and LOGSAFE users that are table owners. The DROP USER command with the CASCADE option has the power to wipe out everything in the database, if used incorrectly. Carefully enter the following commands as the Unix user <oradba>: *<cd avail_filesystem>* *<sqlplus />* *<set feedback off>* *<set heading off>* *<spool drop_P_users.sql>* *<prompt spool drop_P_users.lis>* *<SELECT DISTINCT* *('DROP USER '||owner||' CASCADE;')* *FROM sys.dba_tables* *WHERE owner NOT IN* *('SYS','SYSTEM','ORADBA','ORACLE','TABLE_MASTER', 'GSORTS','MEPESDBA','NPGDBA', 'LOGADMIN')* *AND tablespace_name IN ('JEPES_DATA','LOGSAFE'); >* *<spool off>* *<exit>* Compare each username generated in drop_P_users.sql with the usernames in avail_filesystem/P_JEPES_users.lis and avail_filesystem/P_LOGSAFE_users.lis. Only proceed to Step 18d, after determining that all users to be dropped were JEPES or LOGSAFE users. Do not DROP CASCADE other users. If necessary, 'vi' the file drop_P_users.sql to remove DROP USER statements for users that should not be dropped.	The scripts provided to restore supplementary user backups begin the restoration process by dropping users that do not own database objects (preventing inadvertent loss of critical database objects). When importing a database containing application users that own tables, these users must first be dropped using the powerful 'DROP USER CASCADE' command. To avoid mistakes that may require falling back to the full database import, make sure that all users generated in the drop_P_users.sql script are positively identified as non-critical JEPES or LOGSAFE users.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
18d	Restore Local Oracle User Accounts, If Desired, Step E.	R	After verifying that all users to DROP CASCADE were JEPES or LOGSAFE users (previous step), run the generated script as the Unix user <oradba>: *<sqlplus />* *<set pages 0>* *<set echo on>* *<set feedback on>* *<@avail_filesystem/drop_P_users >* *<exit>* Review the results file drop_P_users.lis.	

<table>
<tr><th>STEP</th><th>STEP TITLE</th><th>SITE</th><th>ACTIONS & COMMANDS</th><th>EXPLANATION</th></tr>
<tr><td><b>18e</b></td><td>Restore Local Oracle User Accounts, If Desired, Step E.</td><td>R</td><td>Drop PDR users that own private synonyms. Carefully enter the following commands as the Unix user <oradba>:

<i><cd avail_filesystem></i>
<i><sqlplus /></i>
<i><set feedback off></i>
<i><set heading off></i>
<i><spool drop_P_users2.sql></i>
<i><prompt spool drop_P_users2.lis></i>
<i><SELECT DISTINCT</i>
<i>(‘DROP USER ‘||owner||’   CASCADE;’)</i>
<i>FROM sys.dba_synonyms</i>
<i>WHERE owner NOT IN</i>
<i>(‘SYS’,’SYSTEM’,’ORADBA’,’ORACLE’,’TABLE_MASTER’,</i>
<i>’GSORTS’,’MEPESDBA’,’NPGDBA’,’LOGADMIN’,’PUBLIC’</i>
<i>)</i>
<i>AND synonym_name LIKE ‘PDR%’; ></i>
<i><spool off></i>
<i><exit></i>

Review the generated file avail_filesystem/drop_P_users2.sql. If necessary, ‘vi’ the file drop_P_users2.sql to remove DROP USER statements for users that should not be dropped.

After verifying that all users to DROP CASCADE were PDR users (previous step), run the generated script as the Unix user <oradba>:

<i><sqlplus /></i>
<i><set pages 0></i>
<i><set echo on></i>
<i><set feedback on></i>
<i><@avail_filesystem/drop_P_users2 ></i>
<i><exit></i></td><td>The scripts provided to restore supplementary user backups begin the restoration process by dropping users that do not own database objects (preventing inadvertent loss of critical database objects). When importing a database containing application users that own private synonyms, these users must first be dropped using the powerful ‘DROP USER CASCADE’ command.

To avoid mistakes that may require falling back to the full database import, make sure that all users generated in the drop_P_users2.sql script are positively identified as PDR users.</td></tr>
</table>

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
18f			Next, drop user information and restore externally identified local Oracle users. As the Unix user <oradba>: *< cd /h/SMDB/Scripts/SM_bld_tables >* Execute the following scripts at the Unix prompt: *<rstr_users.sh>* Review the list file rstr_users.lis. *<rstr_roles.sh>* Review the list file rstr_roles.lis. *<rstr_grants.sh.>* Review the list file rstr_grants.lis. *<sqlplus />* *<@rstr_synonyms>* *<exit>* Review the list file rstr_aynonyms.lis.	The restore users scripts rebuild local Oracle user accounts, roles, synoyms, and a subset of object grants for externally identified users. Each script drops existing user information before rebuilding users from backup.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
18g	Restore Local Oracle User Accounts, If Desired, Step E.	R	Modify the passwords for table owners in the database (table owner passwords were changed to a temporary password). Pick a new password, and substitute it for new_password in the instructions below. From the Unix <oradba> user: *<sqlplus / >* *<alter user GSORTS identified by new_password;>* *<alter user TABLE_MASTER identified by new_password;>* *<alter user MEPESDBA identified by new_password;>* *<alter user NPGDBA identified by new_password;>* *<alter user SYSTEM identified by new_password;>* *<exit>*	Passwords were changed during the restore process to allow the script rstr_grants.sh to grant object privileges on objects in the owner's schema. **Note:** You will find that object grants are missing for users of some applications (those with tables not owned by GSORTS, TABLE_MASTER, MEPESDBA, or NPGDBA.) Recreate missing user grants by running application-specific user scripts after the database is restored.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
19a	Restore User Functional and Oplan Permissions	R	First, check the date on the file /h/SMDB/Scripts/SM_bld_tables/ bkup_user_perm.dmp. If you wish to keep permissions imported from P and you previously chose not to restore local users from backup (Step 18c through 18d), skip to Step 19b. To replace all existing imported permissions with the permissions in the local user backup, follow the steps below. As the Unix <oradba> user: *<cd /h/SMDB/Scripts/SM_bld_tables>* *<cp rstr_user_perm.par rstr_user_perm.old>* *<chmod 755 rstr_user_perm.par>* *<vi rstr_user_perm.par>* Remove the entry 'USER_OPLAN_PERMISSION,' from the line beginning with the word TABLES=(...). Write the file and quit *<ESC :wq>*. Execute the following script from the Unix. prompt as the user <oradba>: *<rstr_user_perm.sh>* Review the file rstr_user_perm.log for serious errors. Ignore errors caused by referential integrity constraints indicating mismatched user ID's and OPLAN's (i.e., IMP-00019, IMP-00003, ORA-02291). As the Unix user <oradba>: *<mv rstr_user_perm.old rstr_user_perm.par>*	This step is required if Step 18c was performed first. If desired, restore user functional and oplan permissions from Supplementary User Backup. **Note:** The restore local backup parameter file is modified to improve performance during disaster recovery; the table USER_OPLAN_PERMISSION will be generated in the next step, and does not need to be imported.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
19b	Restore User Functional and Oplan Permissions (Continued from 19a)	R	Enter the following commands as the <oradba> user, to regenerate user_oplan_permissions for the OPLAN data imported from P: *<sqlplus / >* *<TRUNCATE TABLE table_master.user_oplan_permission;>* *<exit>* *<cd /h/SMDB/Scripts/dl_files>* *<sqlplus / >* *<@cre_user_op_perm>*	This procedure regenerates user oplan permissions, based on the imported OPLAN's. User synchronization with the recovered OPLAN set is ensured.
20	Local Plan Recovery Cleanup	R	Log into Unix as a user with TDBM privileges. (This user must have the Oracle DBA role.) From the Unix prompt, enter: *<cd /h/SM/Scripts/graphic>* <setenv DISPLAY [terminal id]:0> *<start_ss>* Choose GCCS SYSTEM SERVICES, then Plan Management, then Local Plan Recovery Cleanup. Clean up real world OPLANs first, then exercise OPLANs.	Refer to the *System Services Administrator's Manual.* In some situations, the user may get two pop-up windows. One window will show OPLANs local to the restored site that have not been restored. Do Step 21 to recover OPLANs local to the restored site. The other window will show distributed OPLANs routed to the restored site, but not restored. These OPLANs are resident at sites other than the sending site. Assumption 5 stated that the providing site would contain all distributed plans.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
21	Load Local Plan	R	Log into Unix as a user with TDBM privileges. From the Unix prompt, enter: *<cd /h/SM/Scripts/graphic>* *<start_ss>* Choose GCCS SYSTEM SERVICES, then Plan Management, then Reload Plan	Refer to the *System Services Administrator's Manual.*
22	Selective Site Recovery	R	Log into Unix as a user belonging to the JADMIN group. From the Unix prompt, enter: *<cd /h/SM/Scripts/graphic>* *<start_ss>* Choose GCCS SYSTEM SERVICES, then Plan Management, then Selective Site Data Recovery	Use this step to recover OPLANs identified during Local Plan Recovery Cleanup. Refer to the *System Services Administrator's Manual.*
23	Clean Up User Accounts	R	Enter the following commands as the <oradba> user: <sqlplus /> <revoke DBA from oracle;> *<alter user oradba identified by new password;>* *<exit>*	Restore original privileges to database object owner.
24	Remove Export Dump (.dmp) files.	P R	Enter the following commands at both the providing and receiving sites, as the <oradba> user: *<rm avail_filesystem/P_gccs_full.dmp>*	Remove the export dump files from both sites, to free space. If avail_filesystem is under /oracle/smback, this step allows the redo log backup software to function normally.

STEP	**STEP TITLE**	**SITE**	**ACTIONS & COMMANDS**	**EXPLANATION**
25	Turn on Redo Log Archiving	P R	At both the providing and receiving sites, log in as the <oradba> user. From the Unix prompt: *<sqldba lmode=y>* From the SQLDBA> prompt: *<connect internal>* *<shutdown immediate>* If the database does not come down within several minutes, open a new window, and repeat the steps above, using *<shutdown abort>* instead of the command *<shutdown immediate>*. If *<shutdown abort>* is used, also enter the commands *<startup open>* and *<shutdown normal>* before continuing. *<startup mount restrict>* *<alter database archivelog;>* *<alter database open;>* *<alter system disable restricted session;>* *<exit>* Until automatic backups are enabled, monitor the archived redo log backup area (*<df -a /oracle/smback>)* and perform manual redo log backups, if required. **Note:** If the network was brought to Zero Flow during the recovery, or if the operational situation dictates, notify users that the database is available.	Turn on redo log archiving to allow database recovery. Database shutdown/startup brings all rollback segments online.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
26	Verify Distributed OPLAN Synchronization (Required if Step 4c was chosen instead of Zero-flow.)	P S R	Use Plan Network Status Screen in System Services to verify distributed OPLAN synchronization. Choose the OPLANS that are most important to the receiving site. This process may take some time. If significant discrepancies are noted, identify the master site (the one with the most accurate data). Log into Unix as a user with a primary group of JADMIN. From the Unix prompt, enter: *<setenv DISPLAY [terminal id]:0>* *<cd /h/SM/Scripts/graphic>* *<start_ss>* Choose GCCS SYSTEM SERVICES, then PLAN MANAGEMENT, then SELECTIVE SITE DATA RECOVERY. **Note:** Notify local users that the database is available.	Compares high-level ULN, CIN, PIN, and Scheduling and Movement data. Refer to the *System Services Administrator's Manual* Use the Selective Site Data Recovery function to recover lost data.
27	Notification	R	Notify Network TDBM when the database is loaded and GCCS is executing.	Inform the network and local users that the site is available for limited use. The database will execute most actions slower than normal until the daily analyzer is run.
28	Perform Oracle Database Full Online Backup	R	Initiate a manual full database backup. Login as the Unix <oradba> user from the GCCS globe. Enter the command *<br_main>* to open the backup and recovery main menu. Select <F> for FULL BACKUP.	Immediately perform a full database backup, to allow recovery from local tapes if problems are encountered.
29	Reenable Oracle Database Automatic Backups	R	Re-enable automatic full, cumulative, and redo log backups. Select <A> for the Automatic Backup Menu, then enable backups by selecting <F>, <C>, and <R>. Exit to the Unix prompt.	Backups were disabled before recovery; reenable automatic backups tailored to the site schedule.

STEP	STEP TITLE	SITE	ACTIONS & COMMANDS	EXPLANATION
30	Run the Daily Analyzer	R	Only after the Full Online Backup has completed (Step 28), run the daily analyzer to generate optimizer statistics. Execute the job in background from the Unix prompt as the <oradba> user: *</h/SMDB/Scripts/SM_bld_tables/ daily_analyze.sh &>* Clean up temporary files used for database recovery from the directory /oracle/smback at both P and R.	Generate estimated optimizer statistics for performance. Statistics were removed from database objects during the export at P. The daily_analyze.sh is used instead of the weekly_analyze.sh to save execution time, at the expense of optimizer accuracy. The weekly_analyze.sh script should be executed as soon as possible.

SECTION C.6 — ADMINISTRATOR QUALIFICATION

C.6.1 OVERVIEW

GCCS Database procedures were designed to comply with the following design assumption: "Each GCCS C/S site will have a site administrator available at all times. Site administrators should be trained on ORACLE7 and SUN Unix operating system backup and recovery procedures. In addition, each site will have network and direct-dial access to an expert administrator."

Backup and recovery on the GCCS C/S uses automated procedures wherever possible, to free the site administrator from unnecessary work. Even with extensive automation, technical knowledge of Sun Unix and ORACLE7 is required to assure recovery from all failure conditions. Site administrator interaction with the Database expert site also requires basic knowledge of operating system and database commands.

C.6.2 RECOMMENDED QUALIFICATIONS

The following recommendation outlines the skills required to perform backup and recovery procedures at each GCCS C/S database site. Paragraph C.6.2.1 defines Unix qualifications, and Paragraph C.6.2.2 defines ORACLE qualifications.

C.6.2.1 Sun Unix Qualifications

GCCS C/S Database site administrators should meet Sun Unix coursework qualifications in Paragraph C.6.2.1.1, or on-the-job training qualifications in Paragraph C.6.2.1.2.

C.6.2.1.1 Coursework Qualification. As a minimum site administrators should have attended the following Sun Educational Services Course or its equivalent:

System Administration Essentials - Upon completion of this course, participants should be able to:

- Identify basic computer concepts
- Understand Sun's client-server environment
- Manage files and directories with basic SunOS commands
- Create and modify files using the vi editor
- Change permissions on files and directories
- Speed operation of frequently used commands with C shell features
- Customize startup files to include shell variables and features
- Use shutdown and startup procedures
- Communicate over the network using mail and talk
- Recognize components of the SunOS file structure

- Understand NFS distributed computing filesystem concepts
- Understand NIS concepts and use NIS commands
- Use system monitoring commands
- Transport binary files using bar and tar.

C.6.2.1.2 On-the-job Qualification. As a minimum, site administrators should have at least three months of full-time Unix system administration or programming experience, and demonstrate the ability to perform all tasks listed in the Sun Educational Services Course described in Paragraph C.6.2.1.1.

C.6.2.2 ORACLE Qualifications

GCCS C/S Database site administrators should meet ORACLE coursework qualifications in Paragraph C.6.2.2.1, or on-the-job training qualifications in Paragraph C.6.2.2.2.

C.6.2.2.1 Coursework Qualification. As a minimum, site administrators should have attended the following ORACLE Education Courses or their equivalent:

Administer the ORACLE7 Database I - Upon completion of this course, participants will be able to install ORACLE and create, start up, and shut down a database. Participants will also be able to manage an ORACLE database and its users.

Administer the ORACLE7 Database II - Upon completion of this course, participants will be able to take full advantage of the features of ORACLE. Participants will be able to configure and tune a database for optimal performance and monitor database activities, and effectively manage complex user privileges. Participants will also learn to use ORACLE recovery mechanisms to perform incomplete recovery from media failures.

The minimum prerequisite for **Administer the ORACLE7 Database II** is **Administer the ORACLE7 Database I** and **three months of experience** as a database administrator. Six months of experience is recommended.

C.6.2.2.2 On-the-job Qualification. As a minimum, site administrators should have at least twelve months of full-time ORACLE database administrator experience, and demonstrate the ability to perform all tasks listed in the ORACLE Education Courses described in part B-1) above. At least three months of this full-time ORACLE database administration experience should be with ORACLE7.

C.6.3 COURSE INFORMATION

Additional information and a course catalog may be obtained by contacting ORACLE and Sun educational representatives. ORACLE education may be contacted at (commercial) 1-800-633-0575 and Sun Educational Services may be contacted at (commercial) 1-800-422-8020.